BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: **Cedrick Stanislas Collomb**	
Application No.: 10/580,310	**APPELLANTS' BRIEF ON APPEAL**
Filed: April 17, 2007	
Atty Dkt. SCDY 22.572(100809-00332)	
Examiner: Maurice L. McDowell, Jr.	Brief Filed: December 16, 2009
For: IMAGE RENDERING	

MAIL STOP APPEAL BRIEF -- PATENTS
Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANTS' BRIEF ON APPEAL TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Pursuant to 37 C.F.R. § 41.37, submitted herewith for the above-identified application is Appellants' Appeal Brief, along with the required fee under 37 C.F.R. § 41.20(b). The Director is hereby authorized to charge $540.00 for the required fee and any additional fee that may be required, or credit any overpayments, to Deposit Account No. 50-1290. If such a charge is made, please indicate the Attorney Docket No. 100809-00332 (SCDY 22.572) on the account statement.

I. STATEMENT OF THE REAL PARTY IN INTEREST

The real party in interest to the above-identified application and to this appeal is the assignee, Sony Computer Entertainment Europe, Ltd., by virtue of an assignment recorded on April 17, 2007 in the USPTO, found on reel 019264, frame 0859.

II. STATEMENT OF RELATED CASES

Appellants' legal representative and the Assignee of the above-identified patent application do not know of any prior or pending appeals, interferences under 37 C.F.R. § 41.37(g) or judicial proceedings which may be related to, directly affect or be directly affected by or have a bearing on the Board's decision with respect to the above-identified Appeal.

III. JURISDICTIONAL STATEMENT

The Board has jurisdiction under 35 U.S.C. § 134(a). The Examiner mailed a Final Rejection on June 8, 2009, setting a three-month shortened statutory period for response. A Response was filed by the Appellant on September 8, 2009. The Office responded with an Advisory Action on September 21, 2009. Appellant filed a Notice of Appeal on October 8, 2009.

IV. STATUS OF CLAIMS

Claims 1-9 and 13-15 are pending in the present application. A copy of the appealed claims is included in the Claims Appendix.

V. STATUS OF AMENDMENTS

None of the present claims have been amended or canceled since the issuance of the Office's Final Office Action of June 8, 2009, 2009. Copies of the Final Office Action and the Advisory Action are attached as **Exhibits A and B** respectively in the Evidence Appendix.

VI. SUMMARY OF CLAIMED SUBJECT MATTER

A summary of the invention by way of reference to the drawings and specification for each of the independent claims is provided as follows:

Independent claim 1 recites a method performed by a computer of forming a two dimensional map of a three dimensional environment ([0060, 0066]). A map origin (FIG. 7, 1340; FIG. 10, 1600) is located in the three dimensional environment, where a viewing direction vector (FIG. 6, 1250; [0069]) is defined passing through the map origin (FIG. 7, 1340; [0070]), and a one-to-one correspondence is established between map positions in the map and the directions of vectors passing through the map origin ([0078]). The method includes the steps of associating an environment position in the three dimensional environment (FIG. 10; [0079]) with a folded vector (FIG. 11, 1700) that passes through the map origin (FIG. 11, 1600), where the folded vector lies in a plane (FIG. 13, 1800) that contains both the viewing direction vector and the environment position ([0080-83]). The method also includes the step of forming an angle with the viewing direction vector that is a predetermined function of the angle between the viewing direction vector and a vector between the map origin and the environment position ([0080-83]). An environment position is associated with the map position corresponding to the direction of the folded vector associated with that environment position (FIG. 11; [0080-83]), and the computer properties are derived for a map position from the properties of the corresponding environment position ([0079-85]).

Dependent claim 2 recites the features of claim 1 and further recites that the predetermined function is a multiplication by a predetermined quantity ([0080]).

Dependent claim 3 recites the features of claim 2 and further recites that the predetermined function is a multiplication by 0.5 ([0080]).

Dependent claim 4 recites the features of claim 1 and further recites that the one-to-one correspondence of a map point with the direction of a vector through the map origin represents a projection onto a predetermined plane of a point on the vector which is a predetermined distance from the map origin (FIG. 12; [0081-85]).

Dependent claim 5 recites the features of claim 4 and further recites that the predetermined plane is a plane orthogonal to the viewing direction vector ([0073]).

Dependent claim 6 recites an image rendering method that includes the steps of generating a two dimensional map of a three dimensional environment using the method according to claim 1, where, for a point of interest on an object to be displayed, a reflection vector is derived in dependence on a normal vector at the point of interest and a direction of view ([0066-69]). Additionally, a position is referenced in the two dimensional map using the reflection vector, to detect environmental properties at that map position ([0069]). Also. the appearance of the object is varied at the point of interest in dependence on the detected environmental properties ([0071-72]).

Dependent claim 7 recites the features of claim 6, in which the varying step is performed in dependence on a reflectivity of the object at the point of interest ([0069-70]).

Dependent claim 8 recites the features of claim 6, in which the environmental properties represent lighting properties ([0070]).

Dependent claim 9 recites a computer-readable medium having instructions stored therein which when executed, cause a computer to perform the method according to claim 1 (FIG. 1).

Independent claim 13 recites an apparatus for forming a two dimensional map of a three dimensional environment ([0060, 0066]), there being a map origin (FIG. 7, 1340; FIG. 10, 2600) located in the three dimensional environment, a viewing direction vector (FIG. 6, 1250; [0069]) defined passing through the map origin (FIG. 7, 1340; [0070]), and a one-to-one correspondence between map positions in the map and the directions of vectors passing through the map origin ([0078]). The apparatus comprises means for associating an environment position in the three dimensional environment (FIG. 10; [0079]) with a folded vector (FIG. 11, 1700) that passes through the map origin (FIG. 11, 1600). The folded vector is lying in a plane containing both the viewing direction vector and the environment position ([0080-83]) and forms an angle with the viewing direction vector that is a predetermined function of the angle between the viewing direction vector and a vector between the map origin and the environment position ([0080-83]). The apparatus also includes means for associating an environment position with the map position corresponding to the direction of the folded vector associated with that environment position ([0080-83]). Also included is means for deriving properties for a map position from the properties of the corresponding environment position ([0079-85]).

Dependent claim 14 recites the map generating apparatus according to claim 13 and also includes means for deriving a reflection vector, in respect of a point of interest on an object to be displayed, in dependence on a normal vector at the point of interest and a direction of view ([0066-69]); means for referencing a position in the two dimensional map using the reflection vector, to detect environmental properties at that map position ([0069]); and means for varying the appearance of the object at the point of interest in dependence on the detected environmental properties ([0071-72]).

Dependent claim 15 recites a video game machine comprising apparatus according to claim 13 (FIG. 1).

Although specification citations are given in accordance with C.F.R. 1.192(c), these reference numerals and citations are merely examples of where support may be found in the specification for the terms used in this section of the Brief. There is no intention to suggest in any way that the terms of the claims are limited to the examples in the specification. As demonstrated by the citations above, the claims are fully supported by the specification as required by law. However, it is improper under the law to read limitations from the specification into the claims. Pointing out specification support for the claim terminology as is done here to comply with rule 1.192(c) does not in any way limit the scope of the claims to those examples from which they find support. Nor does this exercise provide a mechanism for circumventing the law precluding reading limitations into the claims from the specification. In

short, the specification citations are not to be construed as claim limitations or in any way used to limit the scope of the claims.

VII. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-9 and 13-15 were rejected under 35 U.S.C. §103(a) as being unpatentable over *Voorhies* (US Patent No. 5,704,024), in view of *Cerny*. (US Patent Pub. 2003/0112238). A copy of the *Voorhies* and *Cerny* documents are attached hereto as **Exhibits C and D**, respectively.

VIII. ARGUMENT

A. LEGAL STANDARDS

Obviousness under 35 U.S.C. §103

Regarding Obviousness,

> Section 103 forbids issuance of a patent when "the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains."

KSR Int'l Co. v. Teleflex Inc., 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations including (1) the scope and content of the prior art, (2) any differences between the claimed subject matter and the prior art, (3) the level of skill in the art, and (4) where in evidence, so-called secondary considerations. *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966). See also *KSR*, 550 U.S. at 407 ("While the sequence of these questions might be reordered in any particular case, the [*Graham*] factors continue to define the inquiry that controls.")

The Supreme Court stated that in cases involving more than the simple substitution of one known element for another or the mere application of a known technique to a piece of prior art ready for the improvement, it will be necessary to "determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue." *Id.* at 417-418. The Court noted that "[t]o facilitate review, this analysis should be made explicit." Id. at 418 (citing In re Kahn, 441 F.3d 977, 988 (Fed. Cir. 2006) ("[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness")).

Rejections based on 35 U.S.C. § 103 must rest on a factual basis. In making such a rejection, the examiner has the initial duty of supplying the requisite factual basis and may not, because of doubts that the invention is patentable, resort to speculation, unfounded assumptions or hindsight reconstruction to supply deficiencies in the factual basis. *In re Warner*, 379 F.2d 1011, 1017 (CCPA 1967).

B. THE CLAIMED INVENTION

The claimed features address image rendering in a virtual environment surrounding a displayed object to contribute to the surface appearance of the object. As discussed above in section VI, independent claim 1 recites a method performed by a computer of forming a two dimensional map of a three dimensional environment ([0060, 0066]). A map origin (FIG. 7, 1340; FIG. 10, 1600) is located in the three dimensional environment, where a viewing direction vector (FIG. 6, 1250; [0069]) is defined passing through the map origin (FIG. 7, 1340; [0070]), and a one-to-one correspondence is established between map positions in the map and the directions of vectors passing through the map origin ([0078]). The method includes the steps of associating an environment position in the three dimensional environment (FIG. 10; [0079]) with a folded vector (FIG. 11, 1700) that passes through the map origin (FIG. 11, 1600), where the folded vector lies in a plane (FIG. 13, 1800) that contains both the viewing direction vector and the environment position ([0080-83]). The method also includes the step of forming an angle with the viewing direction vector that is a predetermined function of the angle between the viewing direction vector and a vector between the map origin and the environment position ([0080-83]). An environment position is associated with the map position corresponding to the direction of the folded vector associated with that environment

position (FIG. 11; [0080-83]), and the computer properties are derived for a map position from the properties of the corresponding environment position ([0079-85]).

Dependent claim 2 recites the features of claim 1 and further recites that the predetermined function is a multiplication by a predetermined quantity ([0080]). Dependent claim 3 recites the features of claim 2 and further recites that the predetermined function is a multiplication by 0.5 ([0080]). Dependent claim 4 recites the features of claim 1 and further recites that the one-to-one correspondence of a map point with the direction of a vector through the map origin represents a projection onto a predetermined plane of a point on the vector which is a predetermined distance from the map origin (FIG. 12; [0081-85]). Dependent claim 5 recites the features of claim 4 and further recites that the predetermined plane is a plane orthogonal to the viewing direction vector ([0073]).

Dependent claim 6 recites an image rendering method that includes the steps of generating a two dimensional map of a three dimensional environment using the method according to claim 1, where, for a point of interest on an object to be displayed, a reflection vector is derived in dependence on a normal vector at the point of interest and a direction of view ([0066-69]). Additionally, a position is referenced in the two dimensional map using the reflection vector, to detect environmental properties at that map position ([0069]). Also. the appearance of the object is varied at the point of interest in dependence on the detected environmental properties ([0071-72]). Dependent claim 7 recites the features of claim 6, in which the varying step is performed in dependence on a reflectivity of the object at the point of interest ([0069-70]). Dependent claim 8 recites the features of claim 6, in which the

environmental properties represent lighting properties ([0070]). Dependent claim 9 recites a computer-readable medium having instructions stored therein which when executed, cause a computer to perform the method according to claim 1 (FIG. 1).

Independent claim 13 recites an apparatus for forming a two dimensional map of a three dimensional environment ([0060, 0066]), there being a map origin (FIG. 7, 1340; FIG. 10, 2600) located in the three dimensional environment, a viewing direction vector (FIG. 6, 1250; [0069]) defined passing through the map origin (FIG. 7, 1340; [0070]), and a one-to-one correspondence between map positions in the map and the directions of vectors passing through the map origin ([0078]). The apparatus comprises means for associating an environment position in the three dimensional environment (FIG. 10; [0079]) with a folded vector (FIG. 11, 1700) that passes through the map origin (FIG. 11, 1600). The folded vector is lying in a plane containing both the viewing direction vector and the environment position ([0080-83]) and forms an angle with the viewing direction vector that is a predetermined function of the angle between the viewing direction vector and a vector between the map origin and the environment position ([0080-83]). The apparatus also includes means for associating an environment position with the map position corresponding to the direction of the folded vector associated with that environment position ([0080-83]). Also included is means for deriving properties for a map position from the properties of the corresponding environment position ([0079-85]).

Dependent claim 14 recites the map generating apparatus according to claim 13 and also includes means for deriving a reflection vector, in respect of a point of interest on an object to be displayed, in dependence on a normal vector at the point of interest and a direction of

view ([0066-69]); means for referencing a position in the two dimensional map using the reflection vector, to detect environmental properties at that map position ([0069]); and means for varying the appearance of the object at the point of interest in dependence on the detected environmental properties ([0071-72]). Dependent claim 15 recites a video game machine comprising apparatus according to claim 13 (FIG. 1).

C. CLAIMS 1-9 AND 13-15 WERE IMPROPERLY REJECTED UNDER 35 U.S.C. §103(A) AS BEING UNPATENTABLE OVER *VOORHIES* (US PATENT NO. 5,704,024), IN VIEW OF *CERNY*. (US PATENT PUB. 2003/0112238), AS THE EXAMINER HAS FAILED TO MAKE A PRIMA FACIE CASE OF OBVIOUSNESS.

As argued previously by Appellant, *Voorhies* discloses a technique for generating reflection vectors to index an existing environment map, such as a known cubic map (col. 6, line 22 – col. 7, line 25). In contrast, the independent claims of the present application relate to a different, earlier, stage in the process – in particular, to the generation of a new type of environment map itself.

Claim 1 (and similarly claim 13) recites "forming a two dimensional map of a three dimensional environment" (i.e. generating an environment map). By contrast, col. 9, lines 24 to 27 *Voorhies*, cited for this feature, states that "the present invention provides a method … for generating reflection vectors which can be unnormalized (i.e. have non-unit lengths) and for using these reflection vectors to index locations on an environment map" (see also col. 8, lines 14-49). Thus, a plain reading of this passage teaches that *Voorhies* is generating a new type of reflection vector, and not a new type of environment map.

This is apparent from FIGs. 3 and 6, as well as col. 10, lines 45-64 of *Voorhies*.

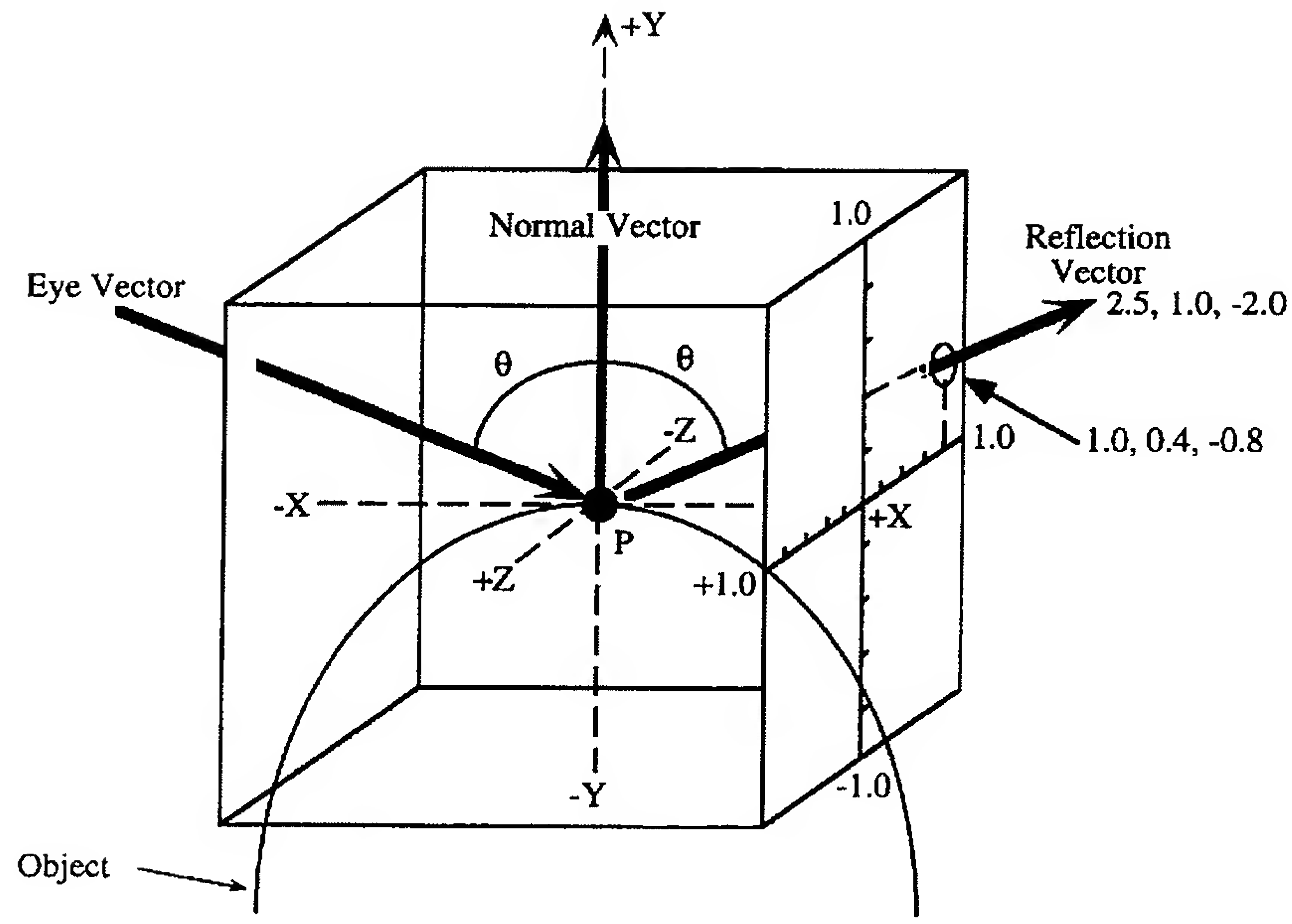


FIG. 6

As can be seen in FIG. 6, *Voorhies* uses a conventional cube mapping arrangement, where six two-dimensional environment maps are positioned around the object (P) so as to cover all possible angles of reflection. This is the 'three dimensional environment map' of *Voorhies*, which is materially different from the claimed configuration.

Appellant submits that an "actual" three dimensional environment and an "environment map of that three dimensional environment" are two different technical concepts. It should be easily appreciated that such a "three dimensional environment map" is not the *actual* three-dimensional environment itself, but a specific representation of it used for the computationally efficient generation of reflections in an object that looks as if they have come from the actual three-dimensional environment. The presently claimed invention makes this distinction between a map and an environment in Claim 1, which recites "forming a two dimensional map of a three dimensional environment."

Turning to FIG. 3 of *Voorhies*, for a given point on the object being rendered, a reflection vector is computed and the appropriate one of the six environment texture maps that

this vector hits is identified (col. 10, line 50). The exact position on this map where the vector hits identifies (indexes) the texture pixel recalled for use in rendering the apparent reflection of the environment at that point on the object (col. 10 lines 57 to 64). This rendering process in *Voorhies* is entirely conventional, in that it computes reflection vectors and retrieves texture information from where that reflection vector points to on an environment map. This feature is concerned with simplifying computation of the reflection vectors themselves (i.e. not normalizing them).

Claim 1 (and similarly claim 13) recites "deriving by the computer properties for a map position from the properties of the corresponding environment position." Thus feature line therefore requires:

i. an environment position;

ii. a map position corresponding to it; and

iii. deriving a property for that map position from the corresponding environment position (e.g. deriving a texture pixel color).

The Examiner cites col. 12, lines 24-31 of *Voorhies*, and states that reflection vectors (from the object being rendered) are used to identify a location on an environment map and retrieve the appropriate shading value from the map. Thus, the Examiner's own reading of *Voorhies* demonstrates a conventional retrieval of texture data from a map at the appropriate reflection point. However, it clearly does not demonstrate deriving initial data for the map (which *Voorhies* then retrieves), and moreover does not mention deriving such data from a corresponding position in the original environment.

Thus the above passages of *Voorhies* cited by the Examiner refer merely to the conventional *use* of environment maps, and not in any way their *generation*. At col. 11, lines 50 to 63, *Voorhies* mentions the entirely conventional generation of two dimensional

environment maps. However, this is the only teaching in *Voorhies* directed to generating such maps.

Since Voohies does not deal with the generation of two-dimensional environment maps, it follows that *Voorhies* does not teach or suggest the following related features found in Claim 1 (and similarly in claim 13):

i. a plane containing both a viewing direction vector and an environment position (i.e. the position currently under analysis) – see plane 1800 of FIG. 13 and the accompanying text of the present application for details;

ii. associating with that environment position a folded vector lying within the above plane - see FIG. 11 and the accompanying text of the present application for details;

iii. the folded vector having an angle that is a function of the angle between the viewing direction vector and a vector between the map origin and the environment position – for details, see the equation at page 15 line 25 of the description (paragraph [0087] in the published application), which halves the angle between the viewing direction vector and a vector between the map origin and the environment position within the above plane;

iv. associating the environment position with the map position corresponding to the folded vector - see FIG. 15 and page 16 lines 10 to 24 (paragraphs [0090-93]), where projection plane 1850 represents the surface of the map being generated; and

v. deriving a property for that map position (based on the folded vector) from the corresponding environment position – see again FIG. 15 and page 16 lines 10 to 24 for details.

Again, an environment map is not the environment itself but a particular and separate representation of that environment; *Voorhies* even makes this distinction at col. 11 lines 50-51 and col. 11 lines 55-58. Claims 1 and 13 also makes the same distinction between environmental maps and the environment itself, reciting "forming a two dimensional map of a three dimensional environment." It is respectfully submitted that it does not make technical sense to equate the three dimensional environment of Claim 1 with maps of a three dimensional environment from *Voorhies*, and in addition it is inconsistent with the teaching of *Voorhies* as outlined above. Hence it also does not make technical sense to apply passages relating to the recall of data from such maps with a claim to the generation of such maps.

In the Final Office Action (paragraph 5), the Examiner argues that *Voorhies* is concerned with generating map properties of a 3D environment, based on col. 11 lines 29-33. However as Appellant has already noted above, it should be clearly understood that a "3D environment map" is for example a cubic arrangement of six environment maps that themselves represent the actual 3D environment, but therefore clearly are not the 3D environment itself. The cited passage clearly relates to locating values in a 3D environment map. It therefore clearly does not teach or suggest generating properties for a map from a 3D environment.

Also in paragraph 7 of the Final Office Action, the Examiner argues that multiplier 1525 is a predetermined function relating the folded vector to other quantities. However, *Voorhies* does not have a folded vector as claimed. Nowhere in *Voorhies* is a folded vector

disclosed with the claimed property of having an angle that is a function of an angle between the viewing direction vector and a vector between a map origin and an environment position; firstly, there are no vectors between the map origin and the environment position because in the cited passage *Voorhies* is using existing environment maps, not the environment itself, as made clear above; secondly there is no plane disclosed containing both a viewing direction vector and an environment position; and thirdly, there is no disclosure of a folding vector lying within that plane. As a result there is no disclosure of the relationship of such a folding vector in such a plane with such other vectors and environment positions. By contrast, it is clear from FIG. 12 of *Voorhies* that multiplier 1525 is applied to reflection vectors between the object and an existing environment map for the purpose of selecting a position on that map.

Throughout the Office Action, Appellants respectfully submit that there appears to be a consistent equivalence given between *generating* properties of an environment map (as in the present invention) and *locating* properties of an existing environment map (as in *Voorhies*). Such a position is improper.

As a further example, the Office Action appears to be equating the "raw" reflection vector (R_x, R_y, R_z) in *Voorhies* (col. 11, lines 41-32) to the "modified" reflection vector that is derived from it (col. 12, lines 24-28 and col. 13, lines 61-65). The Office Action also appears to be equating the "viewing direction vector" of Claim 1 to the Eye vector E in Voories; and the "folded vector" of Claim 1 to the modified reflection vector. This interpretation is simply incorrect.

The modified reflector vector of *Voorhies* is derived as a vector product of E and the normal vector N, followed by some internal manipulation (dividing one or more vector components of the raw reflection vector by numbers derived from other vector components).

In contrast, Claim 1 defines the folded vector as follows:

A. it "passes through the map origin."

B. it is associated with "an environment position in the three dimensional environment" for which Appellant sees no equivalent in *Voorhies* – because *Voorhies* is not concerned with generating map properties in respect of a 3-D environment.

C. it lies in "a plane containing both the viewing direction and the environment position" – because there is no analogy in *Voorhies* to the environment position, this feature is not found in *Voorhies*.

D. it forms "an angle with the viewing direction vector that is a predetermined function of the angle between the viewing direction vector and a vector between the map origin and the environment position" – because there is no environment position defined, this feature is not found in *Voorhies*.

E. it forms an association between a map position and the environment position – again, because there is no environment position defined, this feature is not found in Voohries.

Therefore, it is respectfully submitted that most of the recitation of Claim 1, relating to the definition of the folded vector, is not found in *Voorhies*.

Going further, the Office Action appear to equate the "predetermined function" (a multiplication function in Claim 2) to the multiplier 1325. However, the full recitation of Claim 1 is that the "folded vector forms an angle with the viewing direction vector that is a predetermined function of the angle between the viewing direction vector and a vector between the map origin and the environment position." It is respectfully submitted that the multiplier

1325 merely provides a x2 function in the calculation of the expression R = 2*N*(N·E)-E*(N·N) (*Voorhies* col. 11, line 47), as such it is just a scaling factor in an arbitrarily selected equation – it is not a predetermined function relating the folded vector to the other quantities, as recited in Claim 1.

With respect to Claim 3, contrary to what is alleged in the Office Action that 1320 of FIG. 11 in *Voorhies* is a multiplication by 0.5; it is submitted that 1320 multiples E by (N*N).

In summary, *Voorhies* is concerned with a method of using unnormalized reflection vectors during use of a cubic reflection mapping, and discloses purely conventional methods of generating the environment maps used for such cubic reflection mapping. *Voorhies* fails to teach or suggest, with respect to generating environment maps, at least the aforementioned five features of Claim 1.

Regarding *Cerny*, the document teaches an arrangement for applying (rather than generating) an environment map. FIG. 3 of Cerny shows a map origin (point P 305) and a relationship between a viewing direction (vector e, referred to as an observation vector) and a reflection vector (vector r). That is all. It is respectfully submitted that there is no teaching of the sort that is alleged in section 8 of the Office Action.

The Examiner argues that paragraph [0030] of *Cerny* discloses generating an environment map. Appellant disagrees. *Cerny* is in fact concerned with a different type of environment mapping than *Voorhies* (and the present invention), being concerned with the illumination of objects by virtual light sources, rather than with the reflection of an environment by an object (see paragraph [0005]) of *Cerny*). As a result *Cerny* does not require

an environment map representing the (non-existent) environment as the environment is not reflected.

Rather, as is clear from paragraph [0007] of *Cerny*, a texture map for application to the object itself is being generated based upon how virtual lights illuminate the object. It should be readily appreciated that a texture map of an object is fundamentally different to a two dimensional environment texture map for an environment surrounding the object and which is used to compute how that environment is reflected by the object. Reading paragraph [0030] of *Cerny* in light of paragraph [0007], it is clear that paragraph [0030] relates to generating the texture on the object and is not relevant to the presently claimed invention. Appellant also respectfully asserts that neither paragraph [0030] or any other part of *Cerny* teaches or suggests the aforementioned features of Claim 1 or 13. For the same reasons, there is no apparent reason why one skilled in the art would combine the references in the manner suggested by the Examiner.

For at least these reasons, Appellant submits the rejections to claims 1-9 and 13-15 are improper and should be reversed.

IX. CONCLUSION

In view of the above, Appellant submits that the combination of Voorhies and Cerny fail to teach every element of Claims 1-9 and 13-15 under U.S.C. § 103(a). Additionally, there is no reason why one skilled in the art would combined the documents in the manner suggested by the examiner. Accordingly, reversal of the final rejections, allowance of the rejected claims, and issuance of the subject patent application are respectfully requested.

Appellants' undersigned attorney may be reached by telephone at (212) 940-6311. All correspondence should continue to be directed to our address provided herewith.

Respectfully submitted,

/Pedro C. Fernandez/
Pedro C. Fernandez
Attorney for Appellants
Registration No. 41,741

CUSTOMER NUMBER 026304
Telephone: (212) 940-6311
Fax: (212) 940-8986
Docket No.: SCDY 22.572 (100809-00332)
PCF:fd

CLAIMS APPENDIX

PENDING CLAIMS ON APPEAL OF U.S. PATENT APPLICATION SERIAL NO. 10/580,310

1. A method performed by a computer of forming a two dimensional map of a three dimensional environment, there being a map origin located in the three dimensional environment, a viewing direction vector defined passing through the map origin, and a one-to-one correspondence between map positions in the map and the directions of vectors passing through the map origin;

the method comprising the steps of:

associating by the computer an environment position in the three dimensional environment with a folded vector that passes through the map origin, the folded vector lying in a plane containing both the viewing direction vector and the environment position and forming an angle with the viewing direction vector that is a predetermined function of the angle between the viewing direction vector and a vector between the map origin and the environment position;

associating by the computer an environment position with the map position corresponding to the direction of the folded vector associated with that environment position; and

deriving by the computer properties for a map position from the properties of the corresponding environment position.

2. A method according to claim 1, in which the predetermined function is a multiplication by a predetermined quantity.

3. A method according to claim 2, in which the predetermined function is a multiplication by 0.5.

4. A method according to claim 1, in which the one-to-one correspondence of a map point with the direction of a vector through the map origin represents a projection onto a predetermined plane of a point on the vector which is a predetermined distance from the map origin.

5. A method according to claim 4, in which the predetermined plane is a plane orthogonal to the viewing direction vector.

6. An image rendering method comprising the steps of:

generating a two dimensional map of a three dimensional environment using a method according to claim 1;

for a point of interest on an object to be displayed, deriving a reflection vector in dependence on a normal vector at the point of interest and a direction of view;

referencing a position in the two dimensional map using the reflection vector, to detect environmental properties at that map position; and

varying the appearance of the object at the point of interest in dependence on the detected environmental properties.

7. A method according to claim 6, in which the varying step is performed in dependence on a reflectivity of the object at the point of interest.

8. A method according to claim 6, in which the environmental properties represent lighting properties.

9. A computer-readable medium having instructions stored therein which when executed, cause a computer to perform the method according to claim 1.

10.-12. (Cancelled)

13. Apparatus for forming a two dimensional map of a three dimensional environment, there being a map origin located in the three dimensional environment, a viewing direction vector defined passing through the map origin, and a one-to-one correspondence between map positions in the map and the directions of vectors passing through the map origin; the apparatus comprising:

means for associating an environment position in the three dimensional environment with a folded vector that passes through the map origin, the folded vector lying in a plane containing both the viewing direction vector and the environment position and forming an angle with the viewing direction vector that is a predetermined function of the angle between the viewing direction vector and a vector between the map origin and the environment position;

means for associating an environment position with the map position corresponding to the direction of the folded vector associated with that environment position; and

means for deriving properties for a map position from the properties of the corresponding environment position.

14. An image rendering apparatus comprising:

map generating apparatus according to claim 13;

means for deriving a reflection vector, in respect of a point of interest on an object to be displayed, in dependence on a normal vector at the point of interest and a direction of view;

means for referencing a position in the two dimensional map using the reflection vector, to detect environmental properties at that map position; and

means for varying the appearance of the object at the point of interest in dependence on the detected environmental properties.

15. A video game machine comprising apparatus according to claim 13.

EVIDENCE APPENDIX

EXHIBIT A: Final Office Action dated June 8, 2009.

EXHIBIT B: Advisory Action dated September 21, 2009.

EXHIBIT C: Voorheis (US Patent No. 5,704,024), cited by the Examiner in the Final Office Action dated June 8, 2009.

EXHIBIT D: Cerny (US Patent Pub. 2003/0112238), cited by the Examiner in the Final Office Action dated June 8, 2009.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings.

APPENDIX A

Final Office Action dated June 8, 2009.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/580,310	04/17/2007	Cedrick Stanislas Collomb	SCDY 22.572 (100809-00332	3190

26304 7590 06/08/2009

KATTEN MUCHIN ROSENMAN LLP
575 MADISON AVENUE
NEW YORK, NY 10022-2585

JUN 11 2009

EXAMINER	
MCDOWELL, JR, MAURICE L	

ART UNIT	PAPER NUMBER
2628	

MAIL DATE	DELIVERY MODE
06/08/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

DOCKETED

JUN 11 2009

DATE 9/08/09

10/08/09

11/08/09

12/08/09

Final O/A

Office Action Summary	Application No. 10/580,310	Applicant(s) COLLOMB, CEDRICK STANISLAS
	Examiner MAURICE MCDOWELL, JR	Art Unit 2628

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 April 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-9 and 13-15* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-9, 13-15* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

1. Applicant's arguments filed 3/23/2009 have been fully considered but they are not persuasive.

2. Applicant argues: Initially, at a basic level, Voorhies discloses a technique for generating reflection vectors to index an existing environment map, such as a known cubic map. The independent claims of the present application relate to a different, earlier, stage in the process - in particular, to the generation of a new type of environment map itself. Therefore, Voorhies, at least, does not disclose the first line of Claim 1, nor does it disclose the last two lines of Claim 1.

3. Examiner argues: Voorhies, does disclose the first line of Claim 1, see col. 9 lines 24-27 (The present invention provides a method and an apparatus for generating reflection vectors without vector normalization and for using these reflection vectors to index a three dimensional environment map) and discloses the last two lines of Claim 1 see col. 12 lines 24-31 (At step 620, the R, and Rz 25 components are divided by the magnitude of the R, component to determine the indexed location on the x equals one face of the environment map (which in this example is at y=0.4 and z=-0.8). The values for the indexed location on the indexed face are then used in the conventional manner to 30 retrieve the appropriate shading values from the indexed face of the map. The indexed location on the selected 2-D map can then be supplied to conventional texture mapping algorithms or devices, which then determine the surface shading attributes 35 for the pixel representing displayed point P on the display device. For example, the texture mapping device could determine the pixel shading attributes by averaging the intensity values within the indexed region of the indexed face of the 2-D map).

4. Applicant argues: The Office Action appears to be equating the "raw" reflection vector (Rx, Ry, Rz) in Voorhies (col. 11, lines 41--32) to the "modified" reflection vector that is derived from it (col. 12, lines 24-28 and col. 13, lines 61-65). The Office Action also appears to be equating the "viewing direction vector" of Claim 1 to the Eye vector E in Voorhies; and the "folded vector" of Claim 1 to the modified reflection vector. Applicant disagrees with these alleged correspondences for the following reasons: The modified reflector vector of Voorhies is derived as a vector product of E and the normal vector N, followed by some internal manipulation (dividing one or more vector components of the raw reflection vector by numbers derived from other vector components). In contrast, Claim 1 defines the folded vector as follows: A. it "passes through the map origin" - this may have an analogy with the cubic and octahedral maps of Voorhies. B. it is associated with "an environment position in the three dimensional environment" for which Applicant sees no equivalent in Voorhies - because Voorhies is not concerned with generating map properties in respect of a 3-D environment. C. it lies in "a plane containing both the viewing direction and the environment position" - because there is no analogy in Voorhies to the environment position, this feature is not found in Voorhies. D. it forms "an angle with the viewing direction vector that is a predetermined function of the angle between the viewing direction vector and a vector between the map origin and the environment position" - because there is no environment position defined, this feature is not found in Voorhies. E. it forms an association between a map position and the environment position -again, because there is no environment position defined, this feature is not found in Voorhies. Therefore, it is respectfully submitted that most of the recitation of Claim 1, relating to the definition of the folded vector, is not found in Voorhies.

5. Examiner argues: Voorhies is concerned with generating map properties in respect of a 3-D environment see col. 11 lines 29-33 (... using these reflection vectors to index locations on a cubic environment map (3D) which is aligned with the coordinate system (2D) which specifies the reflection vector) this argument can apply as well to arguments A-E.

6. Applicant argues: It is respectfully submitted that the multiplier 1325 merely provides a x2 function in the calculation of the expression R = 2*N*(N.E)-E*(N.N) (Voorhies col. 11, line 47), as such it is just a scaling factor in an arbitrarily selected equation - it is not a predetermined function relating the folded vector to the other quantities, as recited in Claim 1.

7. Examiner argues: It is a predetermined function relating the folded vector to the other quantities see fig. 12, 1525 and col. 16 lines 9-20 (These multipliers in turn generate the indexed location on the indexed face of the cubic environment map by multiplying the two minor coordinates by the output of the divider).

8. Applicant argues: With respect to Claim 3, contrary to what is alleged in the Office Action that 1320 of Figure 11 in Voorhies is a multiplication by 0.5; it is submitted that 1320 multiples E by (N'N).

9. Examiner argues: See fig. 12, 1525 col. 16 lines 16-19 (The multipliers could multiply by 1/2 or the divider could divide by 2 yielding the same result).

10. Applicant argues: Looking at Cerny, again this is an arrangement for applying (rather than generating) an environment map. Figure 3 of Cerny shows a map origin (point P 305) and a relationship between a viewing direction (vector e, referred to as an observation vector) and a reflection vector (vector r). That is all.

11. Examiner argues: Cerny does teach generating an environment map, see [0030] (The method then processes the reflection vector r to generate texture coordinates (s, t) for each point P).

Claim Rejections - 35 USC § 103

12. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

13. Claims 1-9, 13-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Voorhies et al. Patent No.: 5,704,024 in view of Cerny et al. Pub. No.: US 2003/0112238 A1.

14. Regarding claim 1, Voorhies teaches: A method performed by a computer of forming a two dimensional map of a three dimensional environment, there being a map origin located in the three dimensional environment, a viewing direction vector defined passing through the map origin, and a one-to-one correspondence between map positions in the map and the directions of vectors passing through the map origin; the method comprising the steps of: associating by the computer an environment position in the three dimensional environment with a folded vector that passes through the map origin, the folded vector lying in a plane containing both the viewing direction vector and the environment position and forming an angle with the viewing direction vector that is a predetermined function of the angle between the viewing direction vector and a vector between the map origin and the environment position (fig. 9) (R is the folded vector

because Rx and Ry components are divided by the magnitude of the sum of the reflection vector components (i.e., divided by 5.5) to determine the indexed location on face four of the environment map, also R passes through the origin and forms an angle with the viewing direction vector E); and deriving by the computer properties for a map position from the properties of the corresponding environment position (fig. 6 see also col. 12 lines 13-31).

15. Voorhies doesn't teach: associating by the computer an environment position with the map position corresponding to the direction of the folded vector associated with that environment position.

16. The analogous prior art Cerny teaches: associating by the computer an environment position with the map position corresponding to the direction of the folded vector associated with that environment position (fig. 3 see also [0030]) for the benefit of to implement a system and method of environment mapping that depends upon an observer's location with respect to an object's location and orientation to generate a more realistic reflection pattern, and that is consistent with results of the direct normal projection method for particular object-observer geometries.

17. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine associating by the computer an environment position with the map position corresponding to the direction of the folded vector associated with that environment position as shown in Cerny with Voorhies for the benefit of to implement a system and method of environment mapping that depends upon an observer's location with respect to an object's location and orientation to generate a more realistic reflection pattern, and that is consistent with results of the direct normal projection method for particular object-observer geometries.

18. Regarding claim 2, Voorhies teaches: A method, in which the predetermined function is a multiplication by a predetermined quantity (fig. 12, 1525 see also col. 16 lines 9-20).

19. Regarding claim 3, Voorhies teaches: A method, in which the predetermined function is a multiplication by 0.5 (fig. 12, 1525 see also col. 16 lines 16-19).

20. Regarding claim 4, Voorhies teaches: A method, in which the one-to-one correspondence of a map point with the direction of a vector through the map origin represents a projection onto a predetermined plane of a point on the vector which is a predetermined distance from the map origin (fig. 6).

21. Regarding claim 5, Voorhies teaches: A method, in which the predetermined plane is a plane orthogonal to the viewing direction vector (fig. 6).

22. Regarding claim 6, Voorhies teaches: An image rendering method comprising the steps of: generating a two dimensional map of a three dimensional environment (fig. 4, 512); for a point of interest on an object to be displayed, deriving a reflection vector in dependence on a normal vector at the point of interest and a direction of view (fig. 4, 510); referencing a position in the two dimensional map using the reflection vector, to detect environmental properties at that map position (fig. 6).

23. Voorhies doesn't teach: varying the appearance of the object at the point of interest in dependence on the detected environmental properties.

24. The analogous prior art Cerny teaches: varying the appearance of the object at the point of interest in dependence on the detected environmental properties (fig. 4, 430) for the benefit of to implement a system and method of environment mapping that depends upon an observer's location with respect to an object's location and orientation to generate a more realistic reflection

pattern, and that is consistent with results of the direct normal projection method for particular object-observer geometries.

25. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine varying the appearance of the object at the point of interest in dependence on the detected environmental properties as shown in Cerny with Voorhies for the benefit of to implement a system and method of environment mapping that depends upon an observer's location with respect to an object's location and orientation to generate a more realistic reflection pattern, and that is consistent with results of the direct normal projection method for particular object-observer geometries.

26. Regarding claim 7, Voorhies doesn't teach: A method, in which the varying step is performed in dependence on a reflectivity of the object at the point of interest.

27. The analogous prior art Cerny teaches: A method, in which the varying step is performed in dependence on a reflectivity of the object at the point of interest (fig. 4, 430) for the benefit of to implement a system and method of environment mapping that depends upon an observer's location with respect to an object's location and orientation to generate a more realistic reflection pattern, and that is consistent with results of the direct normal projection method for particular object-observer geometries.

28. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the varying step is performed in dependence on a reflectivity of the object at the point of interest as shown in Cerny with Voorhies for the benefit of to implement a system and method of environment mapping that depends upon an observer's location with respect to an object's location and orientation to generate a more realistic reflection pattern, and that is

consistent with results of the direct normal projection method for particular object-observer geometries.

29. Regarding claim 8, Voorhies doesn't teach: A method in which the environmental properties represent lighting properties.

30. The analogous prior art Cerny teaches: A method in which the environmental properties represent lighting properties (fig. 4, 415) for the benefit of to implement a system and method of environment mapping that depends upon an observer's location with respect to an object's location and orientation to generate a more realistic reflection pattern, and that is consistent with results of the direct normal projection method for particular object-observer geometries.

31. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the environmental properties represent lighting properties as shown in Cerny with Voorhies for the benefit of to implement a system and method of environment mapping that depends upon an observer's location with respect to an object's location and orientation to generate a more realistic reflection pattern, and that is consistent with results of the direct normal projection method for particular object-observer geometries.

32. Regarding claim 9, Voorhies doesn't teach: A computer-readable medium having instructions stored therein which when executed, cause a computer to perform the method.

33. The analogous prior art Cerny teaches: A computer-readable medium having instructions stored therein which when executed, cause a computer to perform the method (fig. 2, 210) for the benefit of to implement a system and method of environment mapping that depends upon an observer's location with respect to an object's location and orientation to generate a more realistic

reflection pattern, and that is consistent with results of the direct normal projection method for particular object-observer geometries.

34. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine computer-readable medium having instructions stored therein which when executed, cause a computer to perform the method as shown in Cerny with Voorhies for the benefit of to implement a system and method of environment mapping that depends upon an observer's location with respect to an object's location and orientation to generate a more realistic reflection pattern, and that is consistent with results of the direct normal projection method for particular object-observer geometries.

35. Regarding claim 13, Voorhies teaches: Apparatus for forming a two dimensional map of a three dimensional environment, there being a map origin located in the three dimensional environment, a viewing direction vector defined passing through the map origin, and a one-to-one correspondence between map positions in the map and the directions of vectors passing through the map origin; the apparatus comprising: means for associating an environment position in the three dimensional environment with a folded vector that passes through the map origin, the folded vector lying in a plane containing both the viewing direction vector and the environment position and forming an angle with the viewing direction vector that is a predetermined function of the angle between the viewing direction vector and a vector between the map origin and the environment position (fig. 9); and means for deriving properties for a map position from the properties of the corresponding environment position (fig. 6 see also col. 12 lines 13-31).

36. Voorhies doesn't teach: means for associating an environment position with the map position corresponding to the direction of the folded vector associated with that environment position.

37. The analogous prior art Cerny teaches: means for associating an environment position with the map position corresponding to the direction of the folded vector associated with that environment position (fig. 3 see also [0030]) for the benefit of to implement a system and method of environment mapping that depends upon an observer's location with respect to an object's location and orientation to generate a more realistic reflection pattern, and that is consistent with results of the direct normal projection method for particular object-observer geometries.

38. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine means for associating an environment position with the map position corresponding to the direction of the folded vector associated with that environment position as shown in Cerny with Voorhies for the benefit of to implement a system and method of environment mapping that depends upon an observer's location with respect to an object's location and orientation to generate a more realistic reflection pattern, and that is consistent with results of the direct normal projection method for particular object-observer geometries.

39. Regarding claim 14, Voorhies teaches: An image rendering apparatus comprising: map generating apparatus (fig. 3, 410); means for deriving a reflection vector, in respect of a point of interest on an object to be displayed, in dependence on a normal vector at the point of interest and a direction of view (fig. 4, 510); means for referencing a position in the two dimensional map using the reflection vector, to detect environmental properties at that map position (fig. 6).

40. Voorhies doesn't teach: means for varying the appearance of the object at the point of interest in dependence on the detected environmental properties.

41. The analogous prior art Cerny teaches: means for varying the appearance of the object at the point of interest in dependence on the detected environmental properties (fig. 4, 430) for the benefit of to implement a system and method of environment mapping that depends upon an observer's location with respect to an object's location and orientation to generate a more realistic reflection pattern, and that is consistent with results of the direct normal projection method for particular object-observer geometries.

42. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine means for varying the appearance of the object at the point of interest in dependence on the detected environmental properties as shown in Cerny with Voorhies for the benefit of to implement a system and method of environment mapping that depends upon an observer's location with respect to an object's location and orientation to generate a more realistic reflection pattern, and that is consistent with results of the direct normal projection method for particular object-observer geometries.

43. Regarding claim 15, Voorhies doesn't teach: A video game machine comprising apparatus.

44. The analogous prior art Cerny teaches: A video game machine comprising apparatus (fig. 2, 200) for the benefit of to implement a system and method of environment mapping that depends upon an observer's location with respect to an object's location and orientation to generate a more realistic reflection pattern, and that is consistent with results of the direct normal projection method for particular object-observer geometries.

45. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine video game machine comprising apparatus as shown in Cerny with Voorhies for the benefit of to implement a system and method of environment mapping that depends upon an observer's location with respect to an object's location and orientation to generate a more realistic reflection pattern, and that is consistent with results of the direct normal projection method for particular object-observer geometries.

Conclusion

46. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MAURICE MCDOWELL, JR whose telephone number is (571)270-3707. The examiner can normally be reached on Mon-Friday 7:30am - 5:00pm Eastern Time.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Xiao Wu can be reached on 571--272-7761. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MM

/XIAO M. WU/
Supervisory Patent Examiner, Art Unit 2628

APPENDIX B

Advisory Action dated September 21, 2009.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/580,310	04/17/2007	Cedrick Stanislas Collomb	SCDY 22.572 (100809-00332	3190

26304 7590 09/21/2009
KATTEN MUCHIN ROSENMAN LLP
575 MADISON AVENUE
NEW YORK, NY 10022-2585

EXAMINER	
MCDOWELL, JR, MAURICE L	
ART UNIT	**PAPER NUMBER**
2628	

MAIL DATE	DELIVERY MODE
09/21/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

DOCKETED
SEP 24 2009
DATE 10/08/09 Next Due Date
12/08/09 Final Due Date
Advisory o/A

Advisory Action Before the Filing of an Appeal Brief	**Application No.** 10/580,310	**Applicant(s)** COLLOMB, CEDRICK STANISLAS	
	Examiner MAURICE MCDOWELL, JR	**Art Unit** 2628	

--The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

THE REPLY FILED 08 September 2009 FAILS TO PLACE THIS APPLICATION IN CONDITION FOR ALLOWANCE.

1. ☒ The reply was filed after a final rejection, but prior to or on the same day as filing a Notice of Appeal. To avoid abandonment of this application, applicant must timely file one of the following replies: (1) an amendment, affidavit, or other evidence, which places the application in condition for allowance; (2) a Notice of Appeal (with appeal fee) in compliance with 37 CFR 41.31; or (3) a Request for Continued Examination (RCE) in compliance with 37 CFR 1.114. The reply must be filed within one of the following time periods:
   a) ☒ The period for reply expires 3 months from the mailing date of the final rejection.
   b) ☐ The period for reply expires on: (1) the mailing date of this Advisory Action, or (2) the date set forth in the final rejection, whichever is later. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of the final rejection.
   Examiner Note: If box 1 is checked, check either box (a) or (b). ONLY CHECK BOX (b) WHEN THE FIRST REPLY WAS FILED WITHIN TWO MONTHS OF THE FINAL REJECTION. See MPEP 706.07(f).

Extensions of time may be obtained under 37 CFR 1.136(a). The date on which the petition under 37 CFR 1.136(a) and the appropriate extension fee have been filed is the date for purposes of determining the period of extension and the corresponding amount of the fee. The appropriate extension fee under 37 CFR 1.17(a) is calculated from: (1) the expiration date of the shortened statutory period for reply originally set in the final Office action; or (2) as set forth in (b) above, if checked. Any reply received by the Office later than three months after the mailing date of the final rejection, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

NOTICE OF APPEAL

2. ☐ The Notice of Appeal was filed on _____. A brief in compliance with 37 CFR 41.37 must be filed within two months of the date of filing the Notice of Appeal (37 CFR 41.37(a)), or any extension thereof (37 CFR 41.37(e)), to avoid dismissal of the appeal. Since a Notice of Appeal has been filed, any reply must be filed within the time period set forth in 37 CFR 41.37(a).

AMENDMENTS

3. ☐ The proposed amendment(s) filed after a final rejection, but prior to the date of filing a brief, will not be entered because
   (a) ☐ They raise new issues that would require further consideration and/or search (see NOTE below);
   (b) ☐ They raise the issue of new matter (see NOTE below);
   (c) ☐ They are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal; and/or
   (d) ☐ They present additional claims without canceling a corresponding number of finally rejected claims.
   NOTE: _____. (See 37 CFR 1.116 and 41.33(a)).
4. ☐ The amendments are not in compliance with 37 CFR 1.121. See attached Notice of Non-Compliant Amendment (PTOL-324).
5. ☐ Applicant's reply has overcome the following rejection(s): _____.
6. ☐ Newly proposed or amended claim(s) _____ would be allowable if submitted in a separate, timely filed amendment canceling the non-allowable claim(s).
7. ☒ For purposes of appeal, the proposed amendment(s): a) ☐ will not be entered, or b) ☒ will be entered and an explanation of how the new or amended claims would be rejected is provided below or appended.
   The status of the claim(s) is (or will be) as follows:
   Claim(s) allowed: _____.
   Claim(s) objected to: _____.
   Claim(s) rejected: _____.
   Claim(s) withdrawn from consideration: _____.

AFFIDAVIT OR OTHER EVIDENCE

8. ☐ The affidavit or other evidence filed after a final action, but before or on the date of filing a Notice of Appeal will not be entered because applicant failed to provide a showing of good and sufficient reasons why the affidavit or other evidence is necessary and was not earlier presented. See 37 CFR 1.116(e).
9. ☐ The affidavit or other evidence filed after the date of filing a Notice of Appeal, but prior to the date of filing a brief, will not be entered because the affidavit or other evidence failed to overcome all rejections under appeal and/or appellant fails to provide a showing a good and sufficient reasons why it is necessary and was not earlier presented. See 37 CFR 41.33(d)(1).
10. ☐ The affidavit or other evidence is entered. An explanation of the status of the claims after entry is below or attached.

REQUEST FOR RECONSIDERATION/OTHER

11. ☒ The request for reconsideration has been considered but does NOT place the application in condition for allowance because: See Continuation Sheet.
12. ☐ Note the attached Information *Disclosure Statement*(s). (PTO/SB/08) Paper No(s). _____
13. ☐ Other: _____.

/XIAO M. WU/
Supervisory Patent Examiner, Art Unit 2628

Continuation of 11. does NOT place the application in condition for allowance because: Applicant argues: For completeness, we further respond to paragraph 5 of the Office Action. The Examiner will at this point appreciate our position that Voorhies in fact explicitly discloses conventional environment map generation at column 11 lines 50-63, and that other cited passages of Voorhies are not further concerned with this process. Furthermore, we also again point out that an environment map is not the environment itself but a particular and separate representation of that environment. Examiner respectfully disagrees: Voorhies discloses generating an environment map see col. 8 lines 14-17 (The present invention provides a method and an apparatus for generating reflection vectors... for using these reflection vectors to index a three dimensional environment map); Examiner feels that it is reasonable to equate this with generating an environment map.

APPENDIX C

Voorheis (US Patent No. 5,704,024), cited by the Examiner in the Final Office Action dated June 8, 2009.

US005704024A

United States Patent [19]

Voorhies et al.

[11] Patent Number: 5,704,024

[45] Date of Patent: Dec. 30, 1997

[54] METHOD AND AN APPARATUS FOR GENERATING REFLECTION VECTORS WHICH CAN BE UNNORMALIZED AND FOR USING THESE REFLECTION VECTORS TO INDEX LOCATIONS ON AN ENVIRONMENT MAP

[75] Inventors: **Douglas Voorhies**, Menlo Park; **James Foran**, Milpitas, both of Calif.

[73] Assignee: **Silicon Graphics, Inc.**, Mountain View, Calif.

[21] Appl. No.: **504,773**

[22] Filed: **Jul. 20, 1995**

[51] **Int. Cl.**^{6} **G06T 15/50**

[52] **U.S. Cl.** **395/126**

[58] **Field of Search** 395/119, 120, 395/126, 129, 130

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,901,064	2/1990	Deering	395/126
5,063,375	11/1991	Lien et al.	395/126
5,253,339	10/1993	Wells et al.	395/126
5,563,989	10/1996	Billyard	395/126

OTHER PUBLICATIONS

Blinn & Newell, "Texture and Reflection in Computer Generated Images", Communications of the ACM, vol. 19, No. 10, pp. 456–461, 1976.

Haeberli & Segal, "Texture Mapping as a Fundamental Drawing Primitive", Proc. Fourth Eurographics Workshop on Rendering, pp. 259–266, Jun. 1993.

Voorhies & Foran, "Reflection Vector Shading Hardware", Computer Graphics Proceedings, Annual Conference Series SIGGRAPH 94, pp. 163–166, Jul. 24, 1994.

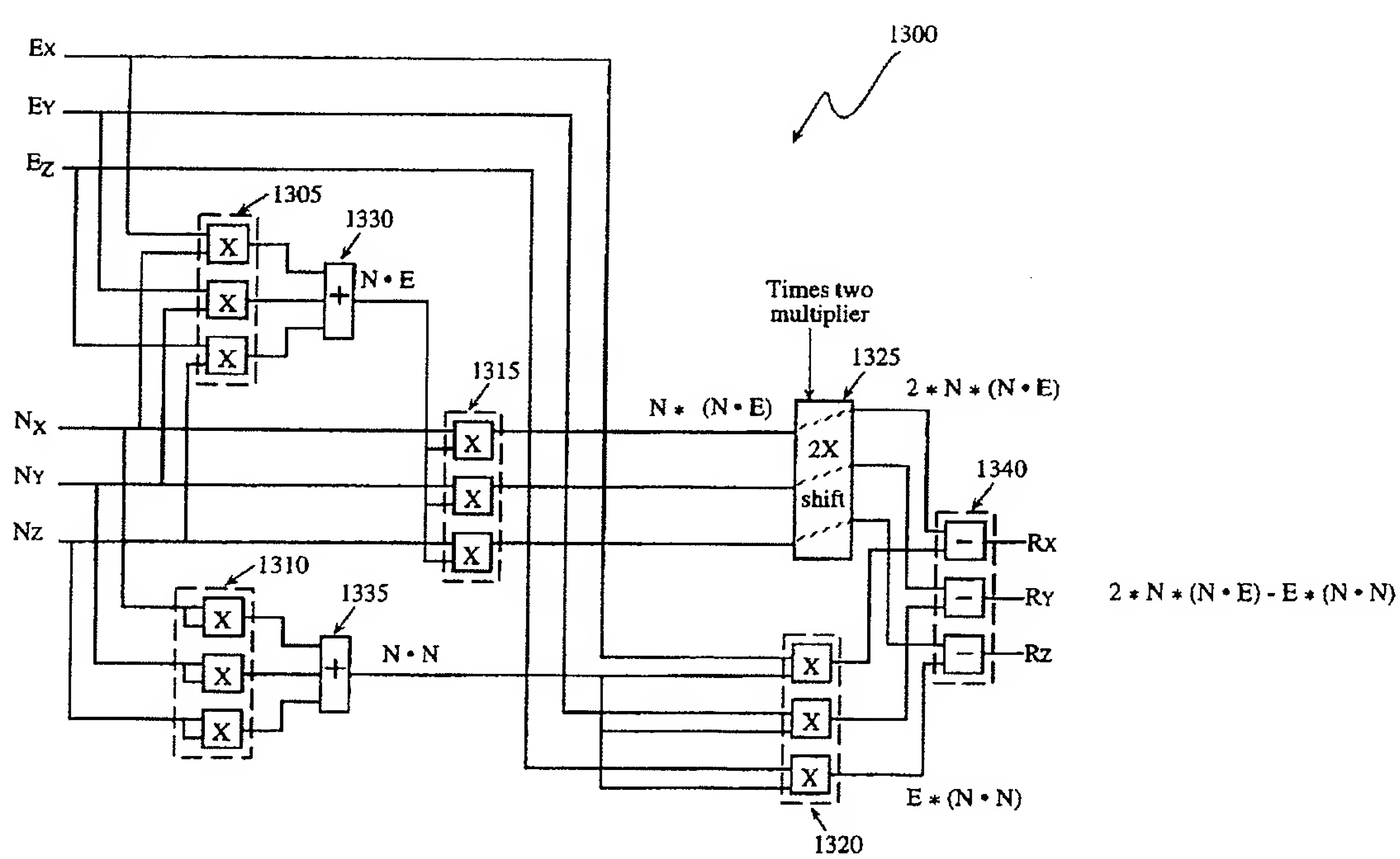
Segal, et al., "Fast Shadows and Lighting Effects Using Texture Mapping", Computer Graphics, vol. 26, pp. 249–252, Jul. 1992.

Primary Examiner—Mark Z. Zimmerman
Assistant Examiner—Cliff N. Vo
Attorney, Agent, or Firm—Blakely, Sokoloff, Taylor & Zafman

[57] **ABSTRACT**

An apparatus for generating a reflection from a three-dimensional environment map. The apparatus includes a reflection vector generator which receives an eye vector and a normal vector neither of which need be normalized. This reflection vector generator generates a reflection vector without vector normalization. The reflection vector generator then couples to a decoder to supply the generated reflection vector. The decoder, in turn, determines a location where the reflection vector indexes the selected 2-dimensional map which forms part of the environment map.

19 Claims, 12 Drawing Sheets



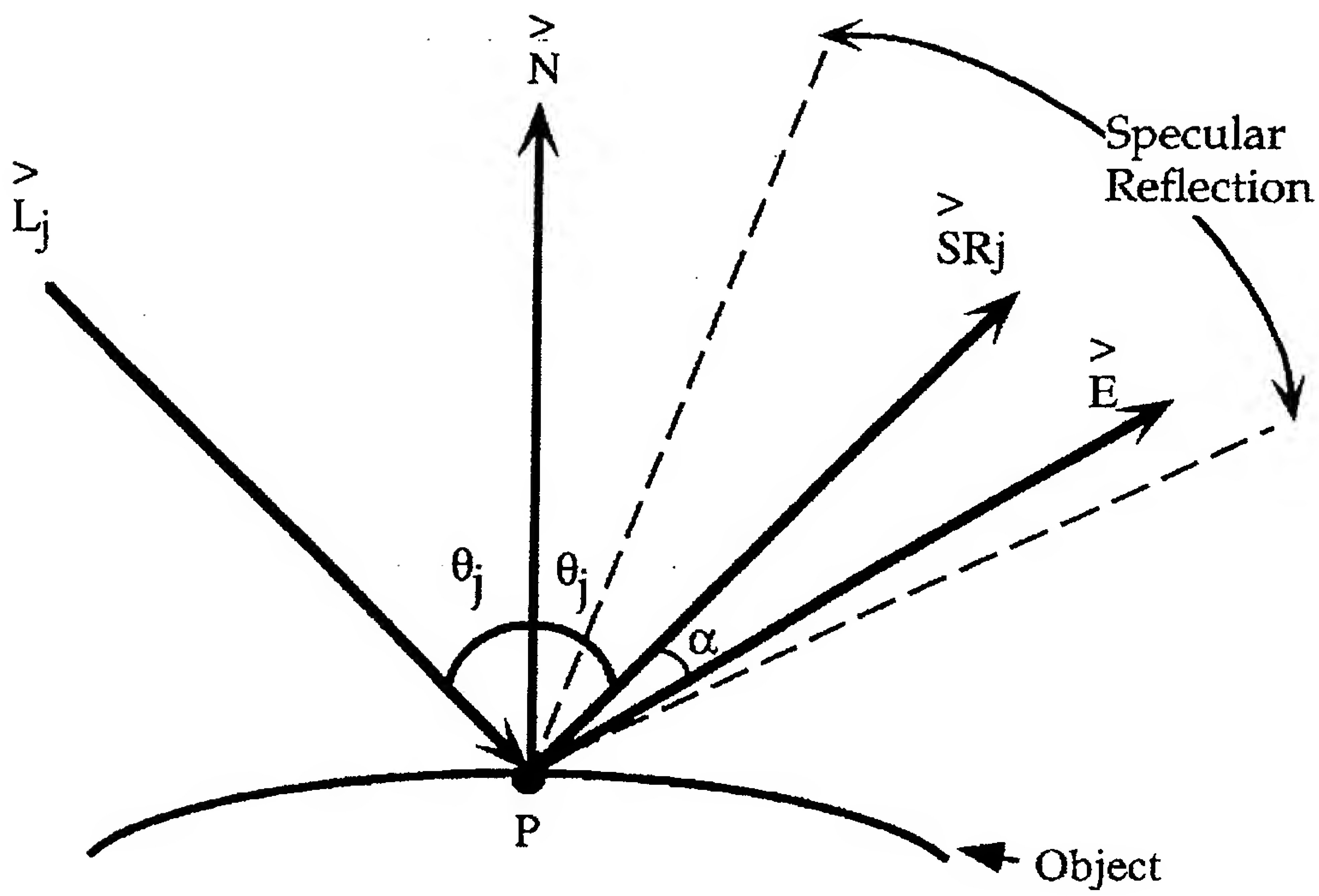


FIG. 1

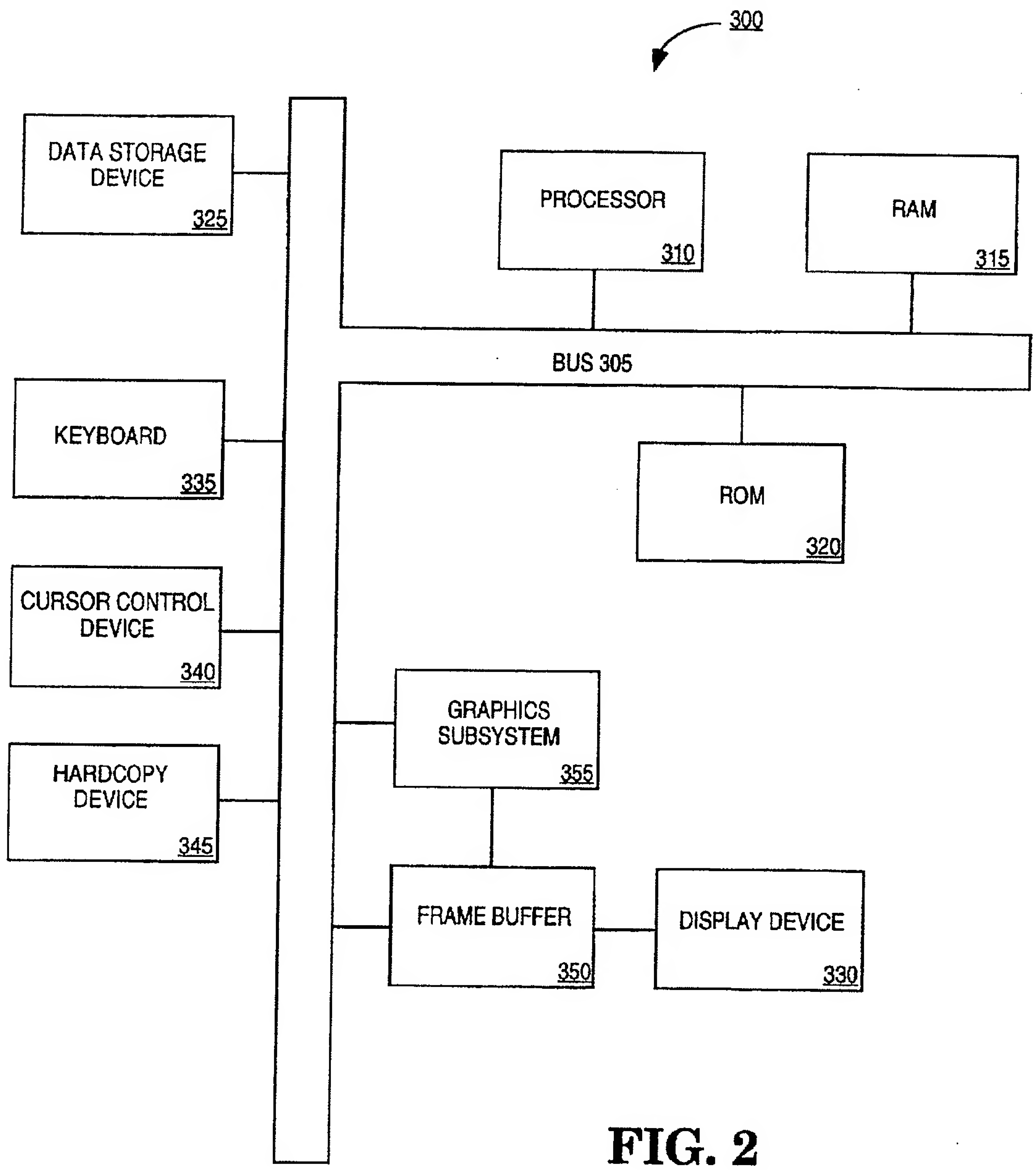

300
DATA STORAGE DEVICE 325
PROCESSOR 310
RAM 315
BUS 305
KEYBOARD 335
ROM 320
CURSOR CONTROL DEVICE 340
GRAPHICS SUBSYSTEM 355
HARDCOPY DEVICE 345
FRAME BUFFER 350
DISPLAY DEVICE 330

FIG. 2

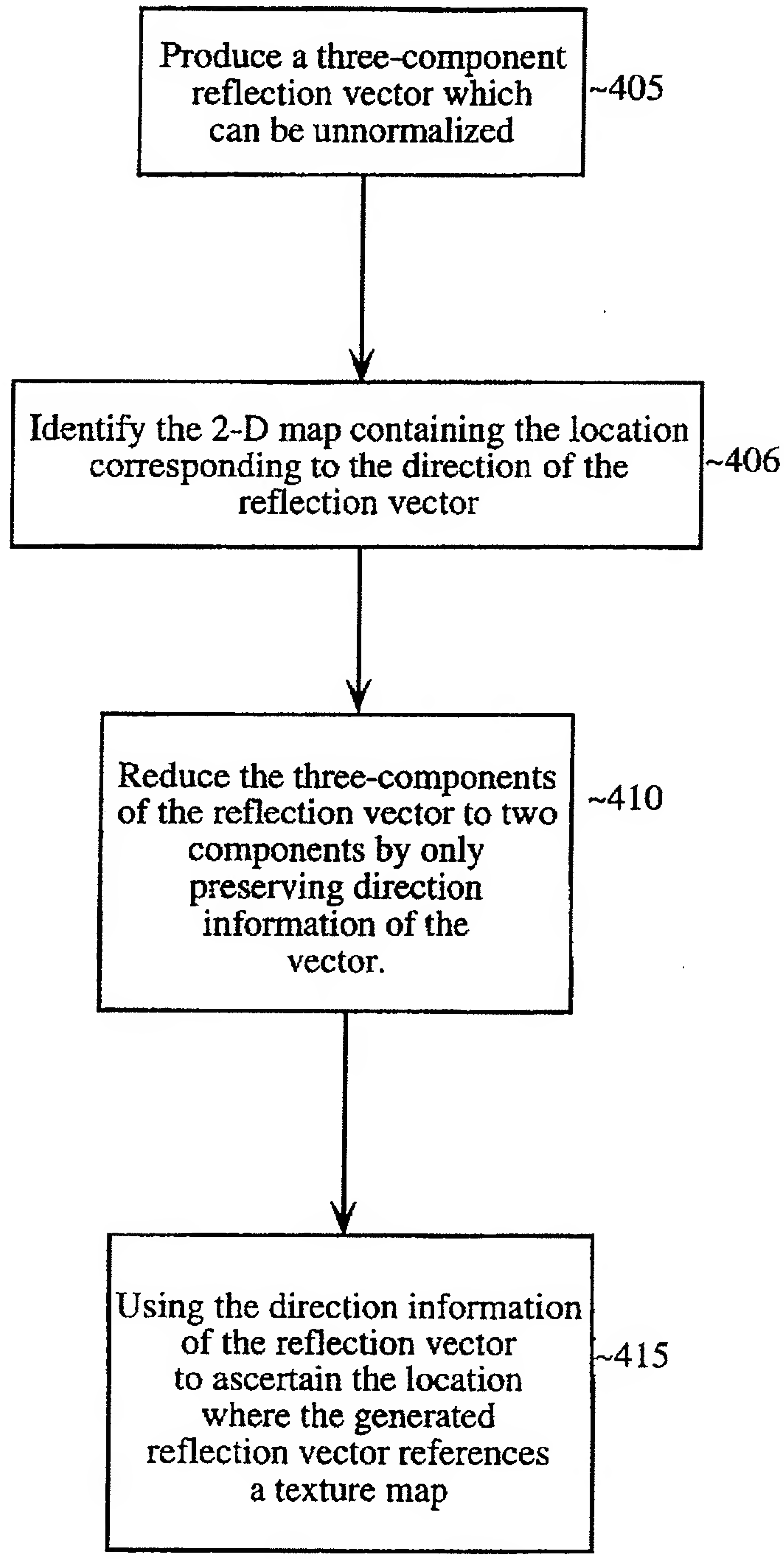

Produce a three-component reflection vector which can be unnormalized
~405
Identify the 2-D map containing the location corresponding to the direction of the reflection vector
~406
Reduce the three-components of the reflection vector to two components by only preserving direction information of the vector.
~410
Using the direction information of the reflection vector to ascertain the location where the generated reflection vector references a texture map
~415

FIG. 3

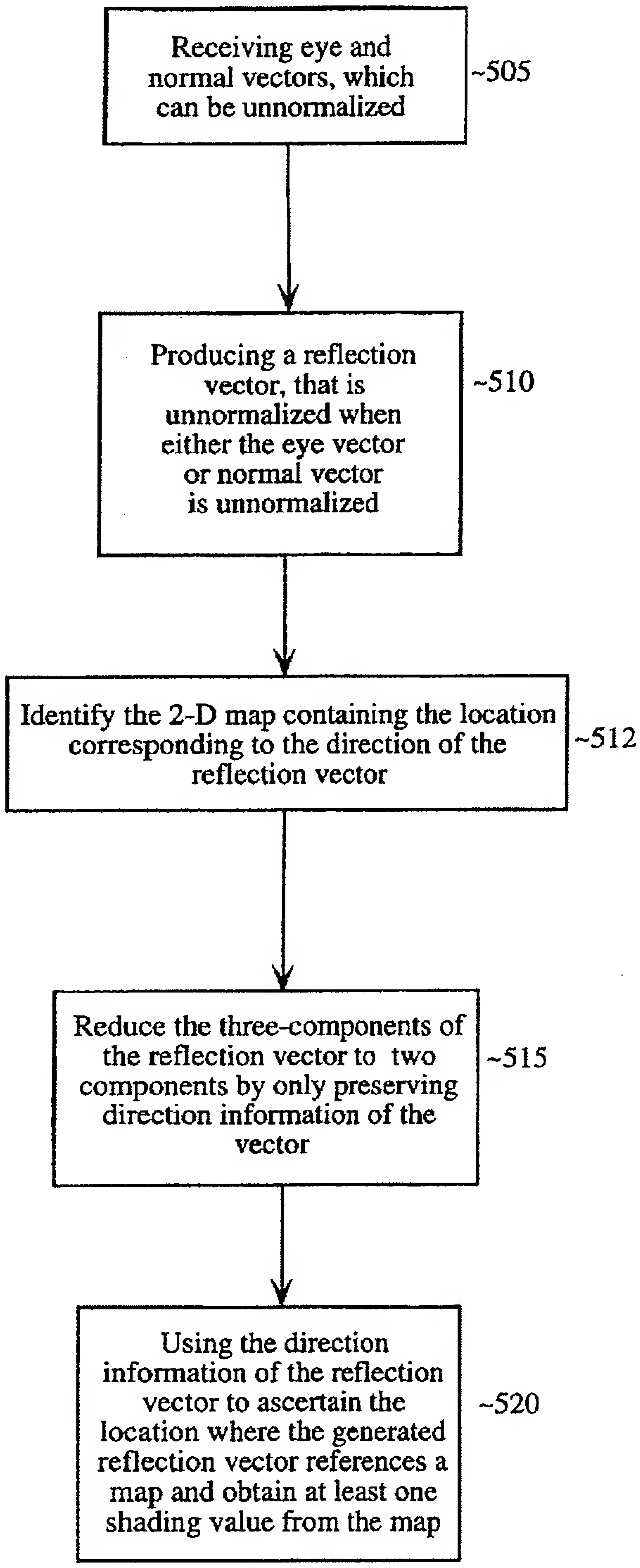

Receiving eye and normal vectors, which can be unnormalized
505
Producing a reflection vector, that is unnormalized when either the eye vector or normal vector is unnormalized
510
Identify the 2-D map containing the location corresponding to the direction of the reflection vector
512
Reduce the three-components of the reflection vector to two components by only preserving direction information of the vector
515
Using the direction information of the reflection vector to ascertain the location where the generated reflection vector references a map and obtain at least one shading value from the map
520

FIG. 4

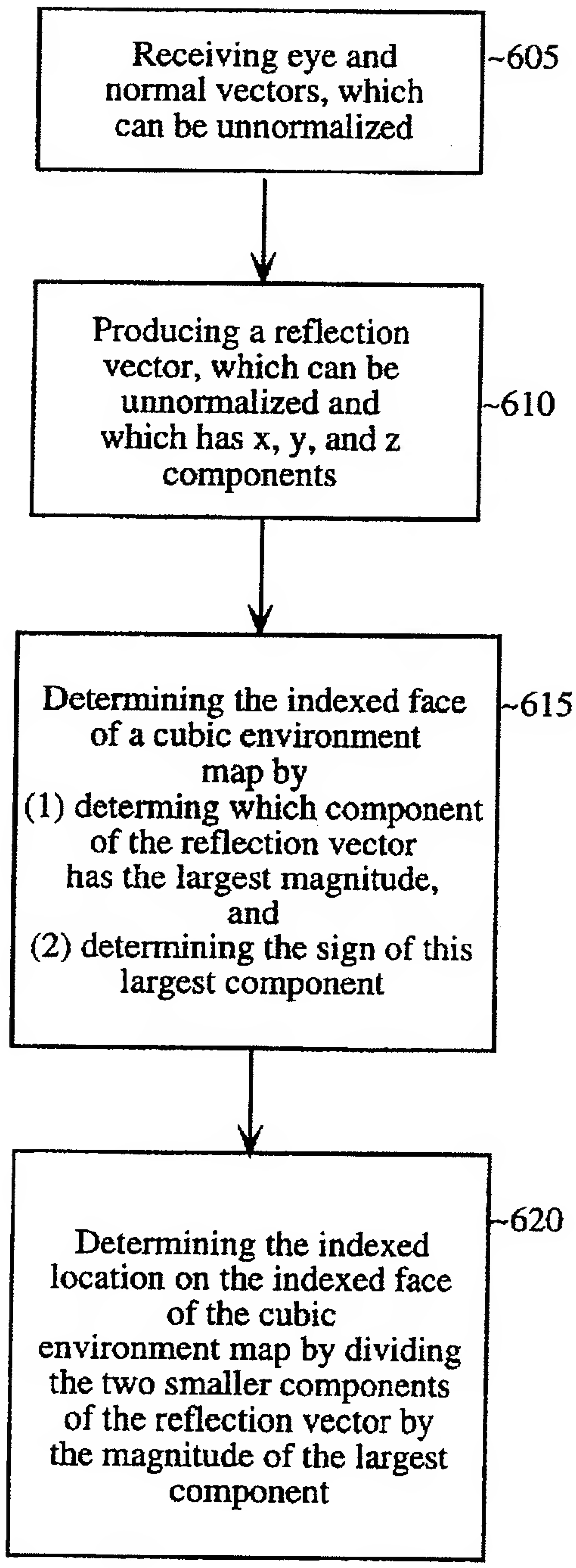

Receiving eye and normal vectors, which can be unnormalized
605
Producing a reflection vector, which can be unnormalized and which has x, y, and z components
610
Determining the indexed face of a cubic environment map by (1) determing which component of the reflection vector has the largest magnitude, and (2) determining the sign of this largest component
615
Determining the indexed location on the indexed face of the cubic environment map by dividing the two smaller components of the reflection vector by the magnitude of the largest component
620

FIG. 5

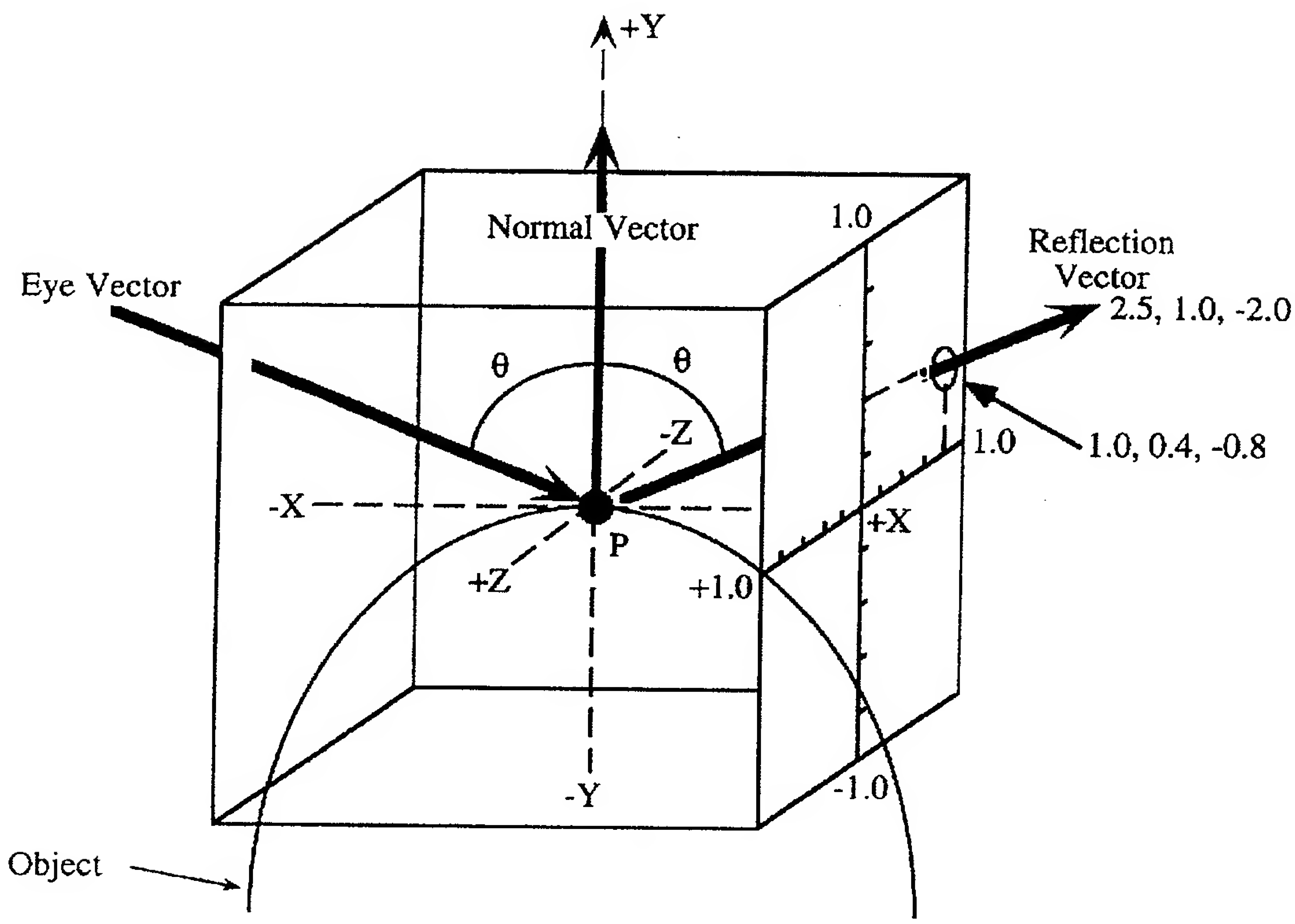

+Y
Normal Vector
1.0
Reflection
Vector
2.5, 1.0, -2.0
Eye Vector
θ
θ
-Z
1.0
1.0, 0.4, -0.8
-X
+X
P
+Z
+1.0
-Y
-1.0
Object

FIG. 6

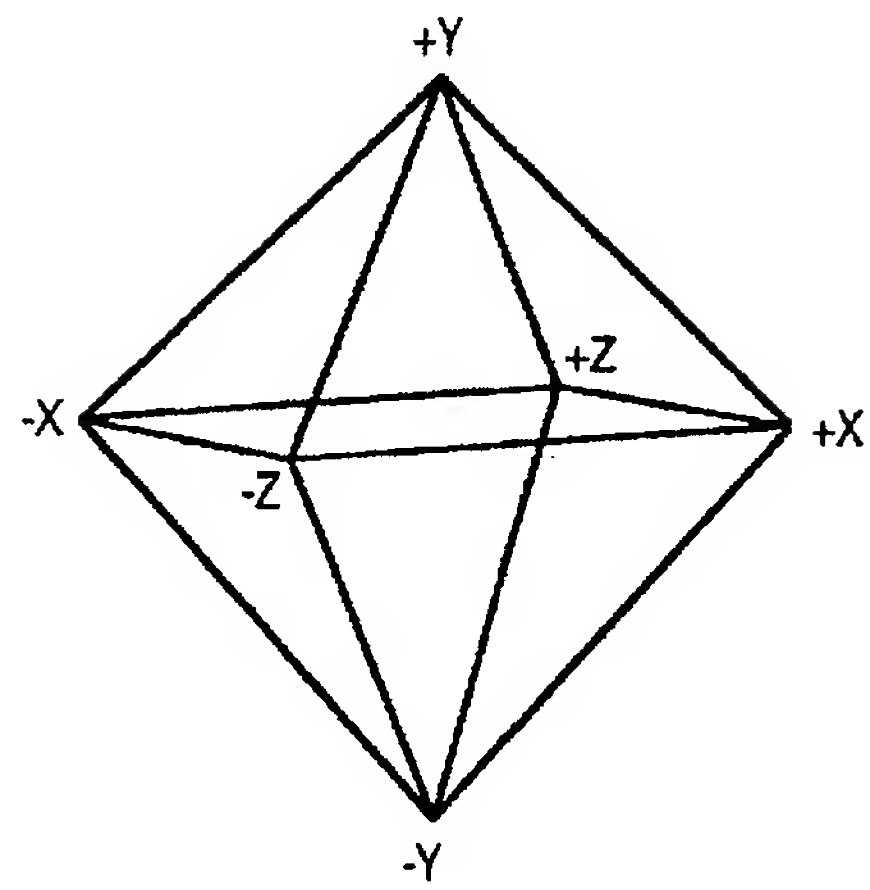


FIG. 7A

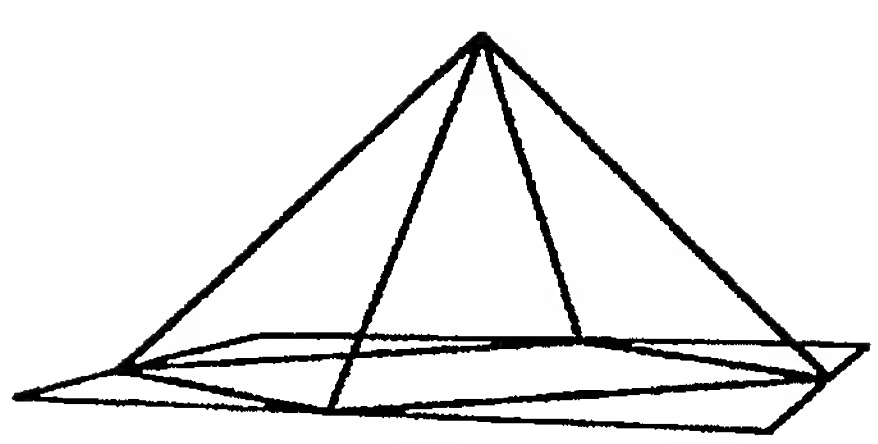


FIG. 7B

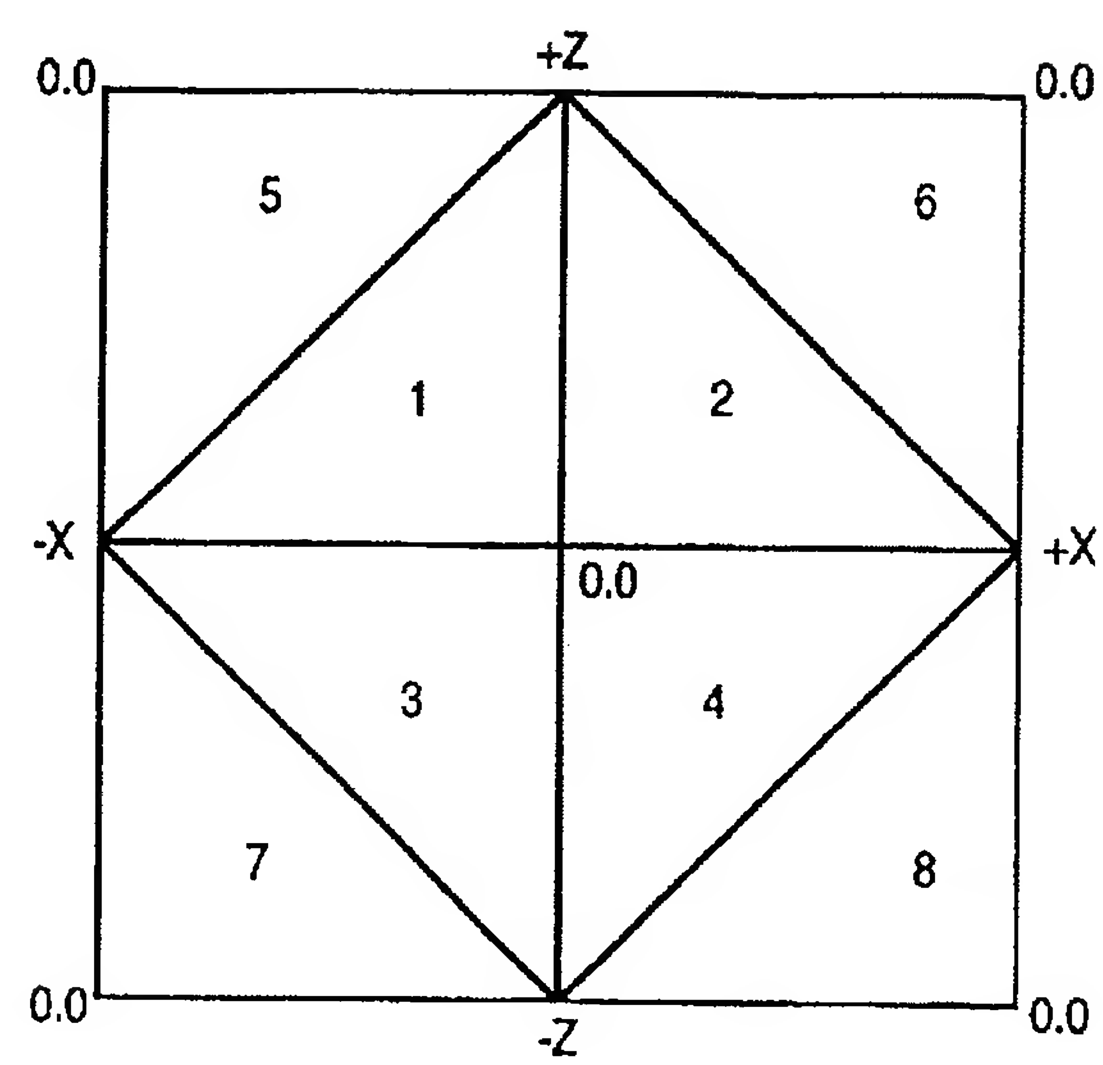


FIG. 7C

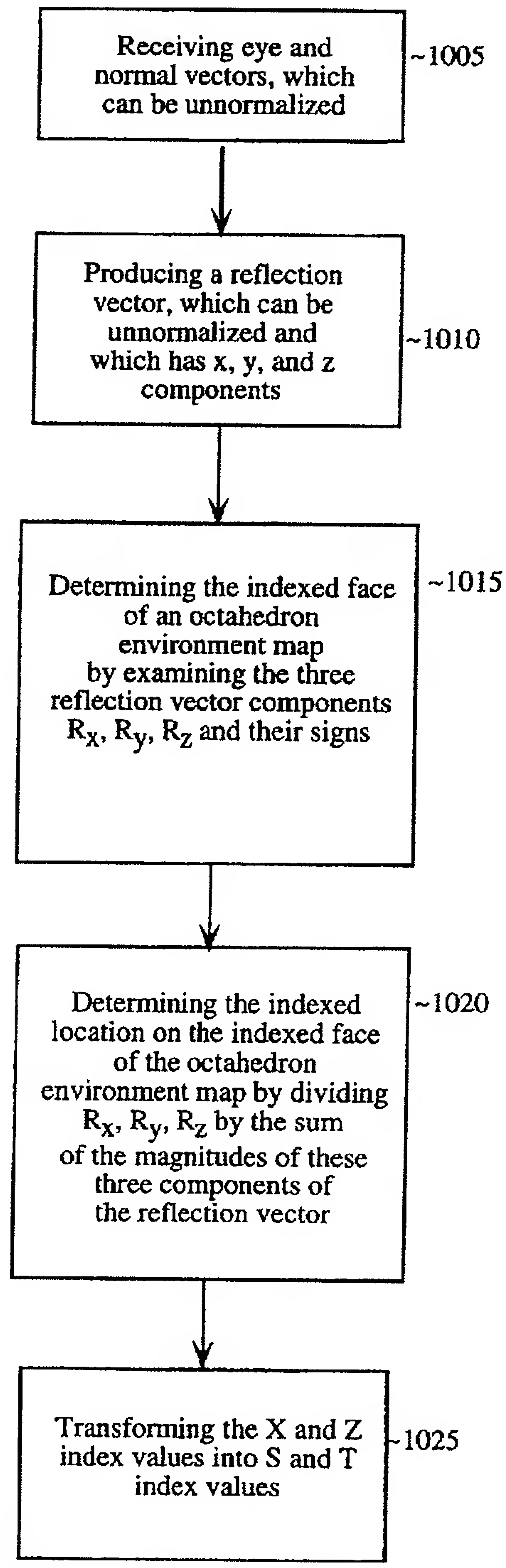

Receiving eye and normal vectors, which can be unnormalized
1005
Producing a reflection vector, which can be unnormalized and which has x, y, and z components
1010
Determining the indexed face of an octahedron environment map by examining the three reflection vector components R_x, R_y, R_z and their signs
1015
Determining the indexed location on the indexed face of the octahedron environment map by dividing R_x, R_y, R_z by the sum of the magnitudes of these three components of the reflection vector
1020
Transforming the X and Z index values into S and T index values
1025

FIG. 8

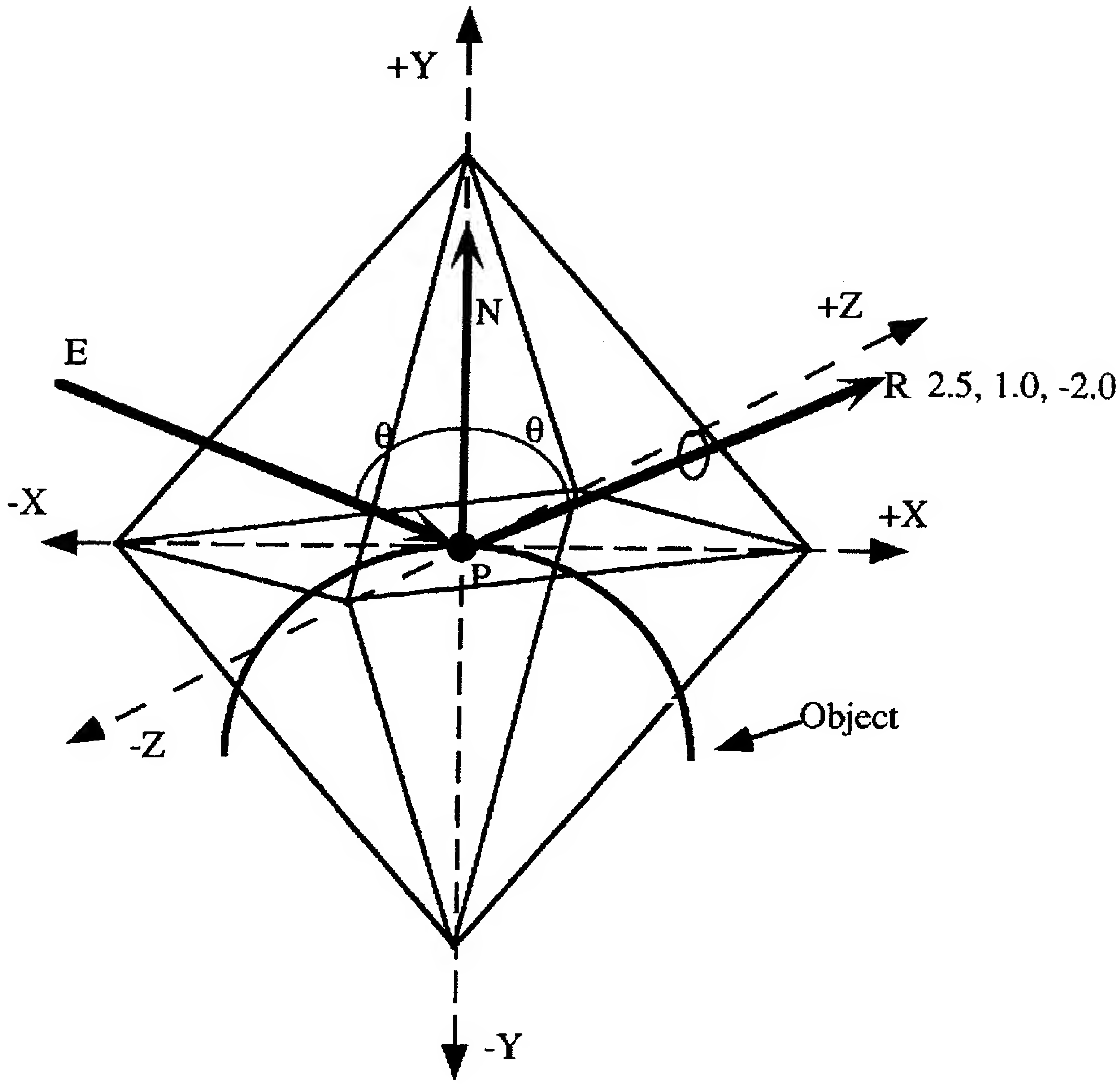


FIG. 9

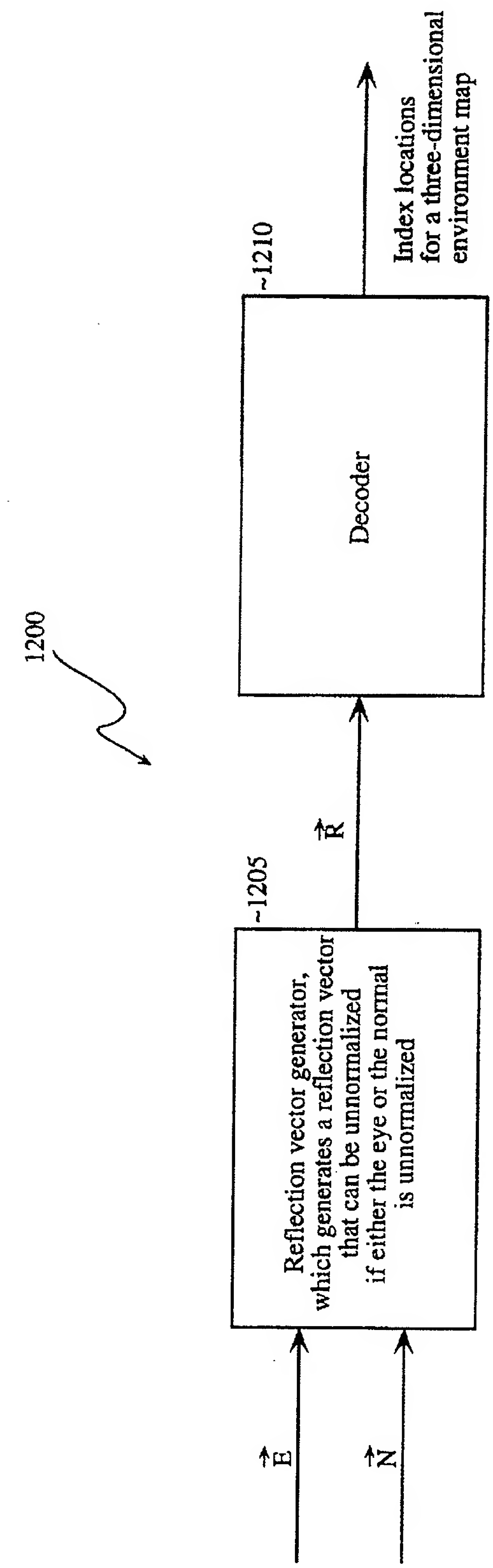

1200
1205
Reflection vector generator, which generates a reflection vector that can be unnormalized if either the eye or the normal is unnormalized
$\vec{E}$
$\vec{N}$
$\vec{R}$
1210
Decoder
Index locations for a three-dimensional environment map

FIG. 10

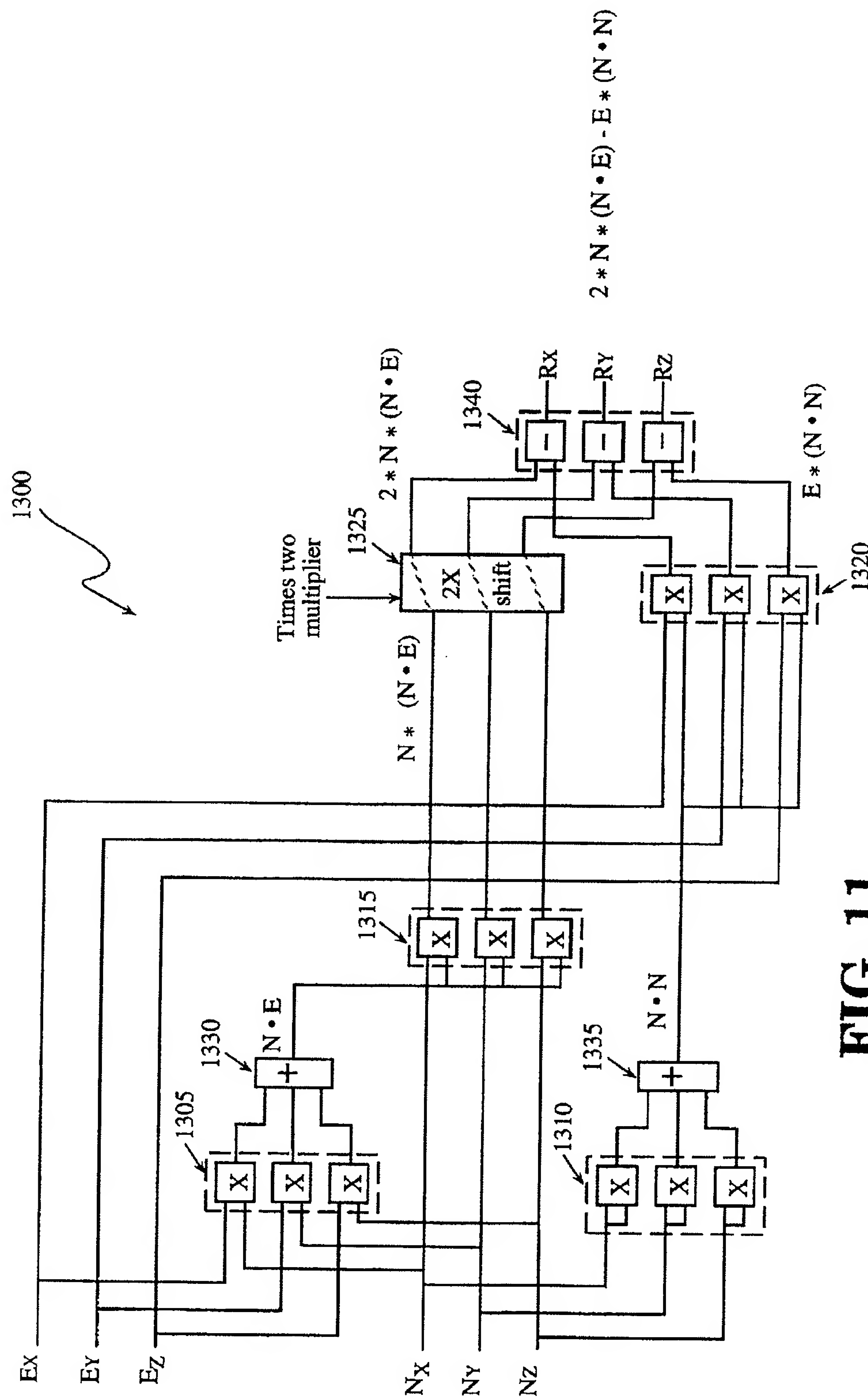

1300
Times two multiplier
1325
2X
shift
1340
1320
1315
1330
1335
1305
1310
N • E
N • N
N * (N • E)
2 * N * (N • E)
E * (N • N)
2 * N * (N • E) - E * (N • N)
RX
RY
RZ
EX
EY
EZ
NX
NY
NZ

FIG. 11

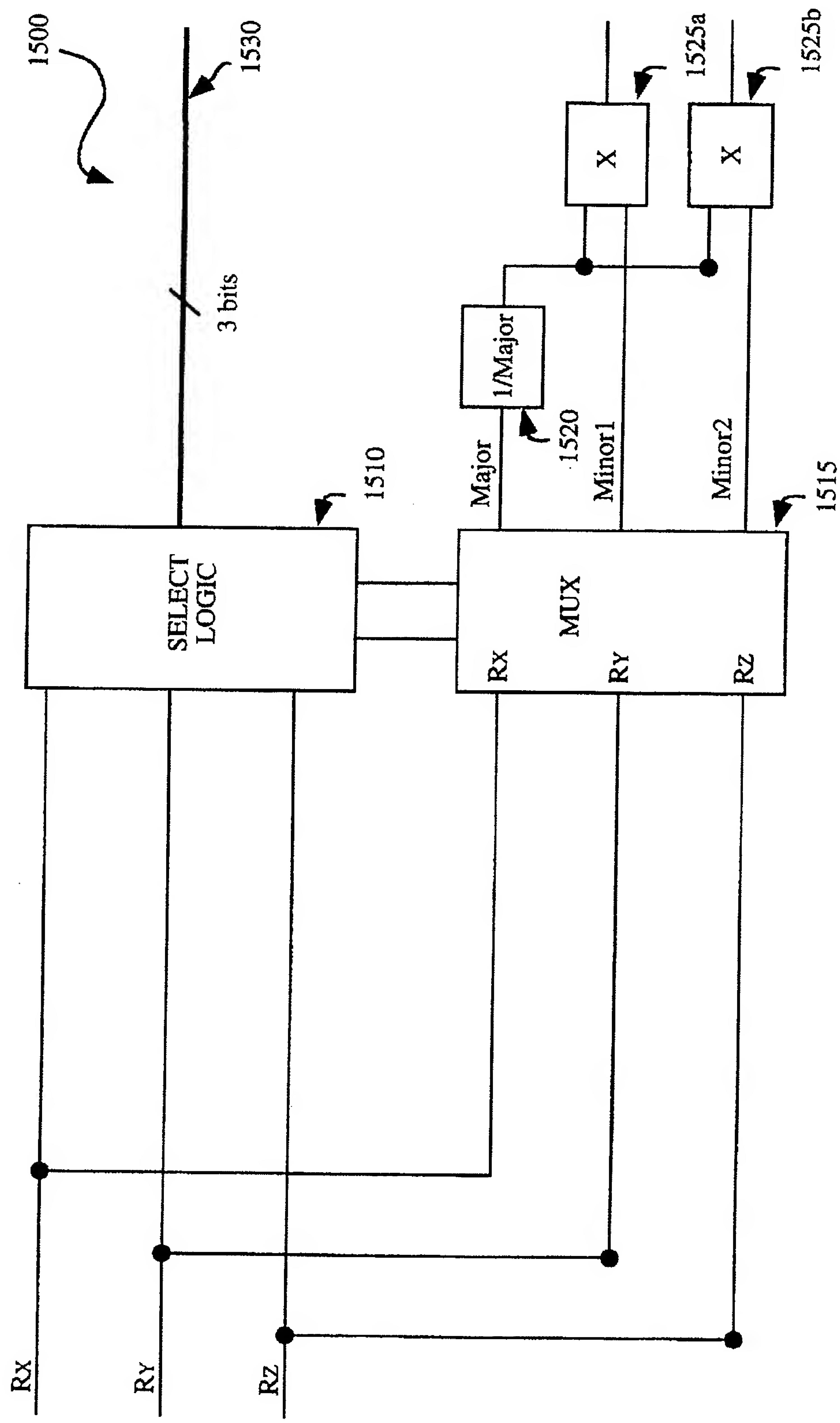

1500
1530
3 bits
1510
SELECT LOGIC
MUX
Rx
RY
Rz
1515
Major
Minor1
Minor2
1/Major
1520
X
X
1525a
1525b
Rx
RY
Rz

FIG. 12

METHOD AND AN APPARATUS FOR GENERATING REFLECTION VECTORS WHICH CAN BE UNNORMALIZED AND FOR USING THESE REFLECTION VECTORS TO INDEX LOCATIONS ON AN ENVIRONMENT MAP

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to the field of computer graphics, and particularly to a method and an apparatus for generating reflection vectors which can be unnormalized and for using these reflection vectors to index locations on an environment map.

2. Description of the Related Art

Interactive computer graphics systems have been widely used in industrial design for the modeling, visualization, and manufacture of complex surfaces such as automobile bodies. A three-dimensional image is represented in a computer graphics system as a collection of three-dimensional geometric primitives (i.e., three dimensional objects), which are defined in a three-dimensional world coordinate system. Each three-dimensional geometric primitive is composed of a number of geometric entities, such as character strings, points, straight lines, curved lines, and filled areas (polygons, circles, etc.). For instance, geometric primitives are commonly represented by polygon meshes, which are sets of connected, polygonally bounded planar surfaces (such as triangles or quadrilaterals).

In addition, often for each graphical primitive, a computer system stores corresponding attributes, which describe how the particular primitive is to be displayed. Common attributes that are stored by computer systems are color specifications, line styles, and text styles. The attributes and the geometric entities of the geometric primitives are then used to render the primitives on a two-dimensional display device, which is composed of a number of pixels whose coordinates are defined relative to a two-dimensional display device coordinate system. This process for transforming a collection of three-dimensional graphical objects into two-dimensional displayed images is called rendering. Literally, the rendering process takes three-dimensional object information and converts it to two-dimensional pixel representation of the object.

One attribute that computer graphic systems have long sought to incorporate into their computer graphic models are surface reflections. Surface reflections are useful for appraising the smoothness and the curvature of complex surfaces as surface reflections are the classic way humans perceive surface shapes. More specifically, because for both real-life and computer generated objects a viewer best observes an object's surface characteristics (i.e., the surface smoothness and curvature) when the object's surface richly reflects light (i.e., the object is shiny) and the viewer moves interactively with respect to the object, computer graphic systems have long sought to produce rich surface reflections at real time speeds (i.e., at speeds that allow the viewer to move interactively with respect to the object).

However, rendering geometric primitives with rich surface reflections at real time speeds has been an elusive goal for reflection rendering hardware. To correctly model the optical physics would require the evaluation of a bi-directional reflectance function at each displayed point of the computer generated model (i.e., at each point of the geometric primitives that is represented by a pixel on the display device). Each bi-directional reflectance function then has to be convolved with the hemispherical illuminating scene visible from the particular surface point corresponding to the reflectance function. Unfortunately, these calculations of the bi-directional reflectance functions and the hemispherical convolutions are computationally unrealizable for real-time reflection rendering systems.

Consequently, prior art computer graphic systems employ gross approximations, such as point light source and mirror reflection approximations, in order to incorporate surface reflections in their models. However, even after using these approximations to model surface reflections, these prior art computer graphic modeling techniques have been unable to render rich surface reflections at real time speeds.

A. Point Light Source Models

Point light source approximation models use a variation of the following illumination equation to represent an object's surface reflection intensity I:

$$I = \sum_{1\le\lambda\le n} I_\lambda = \sum_{1\le\lambda\le n}\left[\left(I_{ambient\lambda} + \sum_{1\le j\le m}\left(I_{diffusej\lambda} + I_{specularj\lambda}\right)\right)\right], \tag{i}$$

where (1) λ represents a wavelength of light, such as red, green, or blue, (2) I_{80} is the intensity of the reflected light, (3) $I_{ambient\lambda}$ is the intensity of the ambient light component, (4) $I_{diffusej\lambda}$ is the intensity of the diffuse reflection light component for each point light source j, and (5) $I_{specularj\lambda}$ is the intensity of the specular reflection component for each point light source j. The ambient light component is used to model reflections of non-directional sources of light which uniformly illuminate a computer generated object. The ambient light component is represented by the following equation:

$$I_{ambient\lambda} = I_{a\lambda} K_a O_{d\lambda},$$

where (1) $I_{a\lambda}$ is the intensity of the ambient light which is constant for all objects, (2) K_a is an object's ambient-reflection coefficient which determines the amount of ambient light reflected from the object's surface, and (3) $O_{d\lambda}$ is the object's diffuse color.

In addition, in order to represent an object's surface reflections due to directional point light sources, computer graphic systems incorporate the diffuse and specular reflection characteristics of an object. FIG. 1 presents a vector diagram of the diffuse and specular reflection characteristics of directional point light sources. The diffuse reflection component ($I_{diffusej\lambda}$) represents a directional point light source's reflections from a surface that are scattered with equal intensity in all directions independent of the viewing direction. In other words, the diffuse reflection component of the reflected light appears equally bright from all the viewing angles and has its brightness dependent only on the angle θ_j between light source j direction vector L_j and surface normal vector N. The diffuse reflection component is represented by the following equation:

$$I_{diffusej\lambda} = I_{pj\lambda} K_d O_{d\lambda} \cos\theta_j,$$

where (1) λ represents a wavelength of light, such as red, green, or blue, (2) $I_{pj\lambda}$ is the intensity of point light source j, (3) K_d is the object's material diffuse-reflection coefficient, (4) $O_{d\lambda}$ is the object's diffuse color, and (5) θ_j represents the

angle between light source j direction vector L_j and surface normal vector N (as shown in FIG. 1). When light source direction vector L_j and surface normal vector N are normalized, cos θ_j equals the dot product between these two vectors.

Prior art computer graphic systems further model the surface reflections due to directional point light sources by incorporating the specular reflection characteristics of an object. The specular reflection component ($I_{specular j\lambda}$) incorporates into the point light source lighting model the highlights (i.e., bright spots) that light sources create on an object. Specular reflections are the result of reflection of incident light in a concentrated region. The specular reflection angle is the angle formed by the specular-reflection vector SR_j and surface normal vector N, and is equal to the angle between light source direction vector L_j and surface normal vector N.

One manner of representing the specular reflection component $I_{specular j\lambda}$ is set forth by the following equation:

$$I_{specular j\lambda}=I_{Pj\lambda}K_sO_{s\lambda}\cos^n\alpha_j,$$

where (1) λ represents a wavelength of light, such as red, green, or blue, (2) $I_{Pj\lambda}$ is the intensity of point light source j, (3) K_s is the object's material specular-reflection coefficient, (4) $O_{s\lambda}$ is the object's specular color, (5) α_j is the angle between the reflection vector SR_j and the viewing direction vector E (as shown in FIG. 1), and (6) n represents the material's specular reflection exponent whose magnitude is proportional to the shininess of the object. If the direction of reflection vector SR_j and viewpoint direction vector E are normalized, then cos α_j equals the dot product of these normalized vectors. Furthermore, reflection vector SR_j can be derived from direction of light vector L_j and surface normal vector N. More specifically, when surface normal vector N and direction of light vector L_j are normalized, the mirroring of L_j about N results in the reflection vector having a value $2*N*(N\cdot L_j)-L_j$. Consequently, if surface normal vector N, direction of light vector L_j, and viewpoint direction vector E are normalized, then

$$\cos\alpha_j=(2*N*(N\cdot L_j)-L_j)\cdot E.$$

One example of a prior art point light source reflection modeling technique shades a polygon surface by linearly interpolating intensity values across the surface of the polygon. More specifically, under this approach, each polygon surface is shaded by performing the following operations: (1) ascertaining the unit normal vector at each polygon vertex, (2) applying a variation of the above-mentioned illumination equation (i) to each vertex to calculate the vertex intensity, and (3) performing Gouraud shading by linearly interpolating the vertex intensities over the surface of the polygon. The Gouraud shading technique is not relatively computationally intensive because it requires normalization operations to be performed only at each vertex. Consequently, this prior art shading model can be used to render specular reflections at real time speeds because it can be implemented by hardware (as it is not computationally intensive).

Unfortunately, there are several disadvantages for rendering specular reflections by using Gouraud shading. For instance, the Gouraud shading technique (i.e., linear intensity interpolation technique) is a poor choice for rendering surface highlights because it accentuates the polygonal nature of the model and introduces Mach banding effects. More specifically, when a curved surface is approximated by a polygon mesh and each polygon facet in the mesh is shaded individually through interpolation, each polygon facet may be easily distinguishable from its neighbors whose orientations are different, because adjacent polygons of different orientation may have different intensity derivatives along their border. When computer graphic systems incorporate Gouraud shading, the lateral inhibition of the receptors in the eye exaggerates the intensity change at any edge where there is a discontinuity in the slope of the intensity (i.e., creates Mach banding effect). Thus, linear intensity interpolation is inappropriate where the surface lighting has high second derivatives (such as near the edge of specular highlights) because such surfaces must either be tessellated so finely that linear intensity interpolation loses its performance advantage or be left more coarsely tessellated and rendered with faceted highlights and obvious Mach bands.

Another prior art point light source reflection modeling technique is the Phong shading model. This shading technique is a more accurate method for rendering surface reflections on a computer generated object, because this shading model linearly interpolates each normal vector at each pixel and then uses these interpolated normal vectors to determine the intensity of the reflected light at each pixel. More specifically, the Phong shading model renders a polygon surface by carrying out the following steps: (1) ascertaining the unit normal vector at each polygon vertex, (2) linearly interpolating the vertex normals over the surface of the polygon, and (3) applying a variation of the above-mentioned illumination equation (i) to calculate projected pixel intensities for surface points. Intensity calculations using interpolated normal vectors produce more accurate results than the direct interpolation of intensity values (as in the Gouraud shading). Consequently, the Phong shading model displays more realistic specular highlights on a polygon surface while reducing the Mach banding effects.

The trade-off, however, is that Phong shading requires extensive calculations per pixel and per light source, which make this shading model too slow for rendering interesting surface reflections in real time. This prevents the use of Phong shading for interactive computer graphics. More specifically, Phong shading is much more computationally intensive than Gouraud shading because Phong shading requires that each interpolated normal vector (N) to be normalized. In turn, the normalization of the normal vector (N) reduces the speed of Phong shading because normalizing a vector involves an inverse square root function, which cannot be inexpensively implemented by hardware.

In addition, the inverse square root function necessary for normalizing a vector is expensive if high precision is required; if this function is approximated by using a table lookup technique any errors will be magnified mercilessly by subsequent specular exponent calculation. Furthermore, in order to perform Phong calculations in hardware, the cosine raised to a power function (which is used for calculating the specular reflection component) is a precision nightmare. Also, performing Phong calculation by hardware is problematic because the $\cos^n\alpha_j$ calculation must be repeated for every light source, which makes the performance of the Phong shading hardware dependent on the number of light sources. This dependence on the number of point light sources is especially problematic when approximating area lights.

B. Mirror Reflection Approximation Models

Two examples of surface reflection modeling techniques that use mirror reflection approximation models are ray

tracing techniques and environment mapping. Ray tracing determines surface reflections by tracing imaginary rays of light from the viewer's eye to the objects in the scene. The basic ray tracing algorithm first sets up a center of projection (i.e., the view from the eye) and a window on an arbitrary view plane. The window is divided into a regular grid whose elements correspond to pixels at the desired resolution. Then, for each pixel in the window, an eye ray is fired from the center of projection through the pixel's center into the scene. Illumination effects accumulated along this ray's path are then assigned to the pixel. The basic ray tracing algorithm also provides for visible-surface detection, shadow effects, transparency, and multiple light source illumination. For example, to calculate shadows, an additional ray is transmitted from the point of intersection to each of the light sources. If one of these shadow rays intersects an object along the way, then the object is in shadow at that point and the shading algorithm ignores the contribution of the shadow ray's light source.

Ray traced displays can be highly realistic, particularly for shiny objects, but they require considerable computation time to generate. More specifically, even if the number of reflections is limited to five or ten reflections, ray tracers have fundamental problems that make them unrealizable for real time rendering hardware. For example, since ray tracing accounts for the local three-dimensional position of reflected objects, ray tracing algorithms are extremely costly and slow. Each reflected ray probes surrounding objects, which multiplies the already high intersection testing load and makes rendering time not linear with complexity.

Surface reflections can also be rendered by environment mapping, which is an alternative procedure for modeling global reflections by (1) defining an array of intensity values that describe the environment surrounding the rendered three-dimensional object, and then (2) projecting these intensity values in relationship to a viewing direction onto the graphical primitives that form the image. Information in an environment map includes intensity values from sources such as light sources, the sky, or other background objects.

An environment map contains a representation of the outside world as viewed from a single point, with the simplifying assumption that the outside world is infinitely far away. Thus an environment map M is indexed only by the reflection direction. This indexing does not communicate the distance between the object and the environment. Furthermore, typically environment mapping algorithms presume that other objects in the three-dimensional image do not obscure the projection of the environment onto a particular object.

To render surface reflections for an object, reflection vectors are first computed for all displayed points of the geometric primitives (i.e., for all geometric primitives' points that are represented by pixels on the display device). A reflection vector can be derived for a particular displayed point P (which is represented by a corresponding pixel on the display device) of a geometric primitive by mirroring the eye (i.e., viewpoint direction) vector E about the surface normal vector N at point P. If surface normal vector N and eye vector E are normalized, the mirroring of E about N yields the following equation for the reflection vector:

$$R_n=2*N_n*(N_n\cdot E_n)-E_n, \qquad \text{(ii)}$$

where the subscript "n" denotes that the vector is normalized.

This reflection vector is then used to index (i.e., to reference) a location on environment map M. Conventional texture mapping techniques then can be employed to impose on the pixel representing displayed point P the two-dimensional image that is obtained at the indexed location in the environment map. For example, the intensity of the reflected light at the pixel corresponding to point P can be determined by averaging the intensity values within the indexed region of the environment map. Anti-aliasing also can be incorporated into the environment mapping model by filtering some number of intensity values surrounding the indexed location in the environment map.

Environment mapping avoids the performance shortcomings of ray tracing because environment mapping makes several approximations. For example, because environment maps represent an outside world at infinity, they are an arbitrary function of direction alone and ignore parallax (i.e., ignore the position of an object relative to the center of the environment in determining the location to sample on the environment map). Ignoring parallax dramatically simplifies the calculation of the index into the map. Thus, the environment map may be pre-computed since it is the image of the environment as viewed from a single reference point.

One implementation of environment mapping in the prior art involves the computation of a normalized eye vector E and a normalized surface normal vector N for each pixel on the object to determine a reflection vector R which is used to index a location on one of 6 faces of a cubic environment map. A cubic environment map can be created by taking six pictures of the environment that surrounds an object with a wide angle camera that has a horizontal and vertical view of 90°. These environment maps are then often stored as 6 2-dimensional maps in texture map storage. This implementation is performed in software and very slowly produces, even on fast hardware, good, distortion-free results. This prior art environment mapping technique is slow because each E and N vector at each displayed pixel must be normalized in order to generate a reflection vector R at each pixel based on the prior art equation (ii), and the normalization process is computationally intensive and hence slow. The results are good because the calculations are performed for all displayed pixels without linear interpolations between vertices of a polygon. Thus, while the results are good, the slowness of this implementation precludes its use for real time, interactive uses of environment mapping.

A faster alternative implementation of environment mapping uses interpolation from the reflection vectors at the vertices of the geometric primitive rather than calculating normalized reflection vectors at each displayed pixel of the primitive. In this approach, described in Haeberli, P. and Segal, M., a Texture Mapping as a Fundamental Drawing Primitive, Proc. Fourth Eurographics Workshop on rendering, Paris, France, June 1993, pp. 259–266, a normalized reflection vector is generated, using the prior art approach of equation (ii) above, at each vertex of a polygon. That is, normalized E and a normalized N vectors are computed only for the vertices of the polygon and then normalized R vectors are computed for only the vertices. These R vectors at the vertices are then used to index the environment map. In this implementation, the environment map is also a cubic environment map, and each reflection vector from the particular vertex indexes a location on one of the six faces of the cubic environment map unless the polygon has reflections onto 2 (or more) different faces of the cubic environment map. If this reflection onto 2 or more faces occurs, then the polygon is subdivided into 2 or more polygons, each of which projects only onto a single face of the cubic environment map, and the reflection vectors for each vertex of each subdivided polygon are used to index

into the selected face of the map. For displayed pixels of the polygon which are not at the vertices, the reflection vector for each such pixel is determined from interpolation (usually linear) of the reflection vectors at the vertices. The use, in this implementation of environment mapping, of interpolation speeds up the processing required to produce an image with reflections from an environment map because normalization, although required for some pixels, is only required for the vertices of displayed polygons. However, the use of interpolation introduces distortion into the reflections produced by this technique; this is because the indexes into the environment map are linearly interpolated between the vertices and linear interpolation does not accurately correspond to the angular change in the reflection vector across a polygon. These distortions can make the image appear less realistic. Another disadvantage of this technique is that subdivision of polygons is necessary when the reflections project onto 2 or more faces; this subdivision also requires computation time and complicates the environment mapping process. Thus while this technique requires less normalization and is faster than the first environment mapping technique described above and allows the viewer to be movable, this technique suffers from distortion in the reflections (producing distorted images) and requires polygon subdivision.

Yet another implementation of environment mapping uses a circular environment map and has been descried in Haeberli and Segal, supra at page 264. Also see, e.g. FIG. 1 at page 164 in Voorhies, D. and Foran, J., Reflection Vector Shading Hardware, Computer Graphics Proceedings, SIGGRAPH 94, pages 163–166 (1994). In this implementation, a single circular environment map is indexed by a normalized reflection vector at each vertex of the polygon, and the reflection vector indices for the remaining pixels of the polygon are interpolated between the indexes at the vertices of the polygon. Since there is only one map, a polygon's reflections cannot project onto different faces and consequently polygon subdivision is not necessary; this improvement makes the process faster and less complicated. While some normalization is required, it is not done for all pixels and thus this implementation is relatively fast; in fact, it is fast enough to be considered practical for some real time, interactive computer graphics.

Unfortunately this implementation with a circular map has several disadvantages. First, some normalization is still required. Furthermore, indexing a circular texture map is correct only for rendering with a fixed viewer. This latter limitation is a poor match to emulating the physical inspection of a car body, for example, where the viewer walks around a stationary car. Virtual reality devises such as the stereoscopic boom emphasize this natural "inspection" paradigm, where the object and environment stay fixed, while the viewer roams at will. Altering the eye position requires creation of a new map, which is challenging to do in real time. Thus, this implementation is not practical for most real time, interactive computer graphics.

Interpolation within a circular map also has two severe artifacts. Grazing reflections map to points near the perimeter, but they are extremely sensitive to object normals. A tiny object rotation can result in a vertex texture index snapping to the opposite side of the map. Secondly, linear texture index interpolation in this "reflection-in-a-sphere" circular map causes severe distortion of the reflection, especially for near-grazing reflections. Although there is sufficient resolution and minimal distortion at the center of the map, the periphery may be very distorted and the mapping extremely anisotropic.

It can be seen from this discussion that while some implementations of environment mapping yield good, distortion-free results but are slow, other implementations are faster, allowing real-time interactivity, but at the expense of distorted reflections and, in the case of circular environment maps, a fixed viewer, and, in the case of cubic environment maps with interpolation, polygon subdivision. It will be appreciated that it is desirable to provide computer graphics having fast, real-time, interactive reflection shading of objects without distortion, without polygon subdivision and with a moveable viewer.

SUMMARY OF THE INVENTION

The present invention provides a method and an apparatus for generating reflection vectors without vector normalization and for using these reflection vectors to index a three dimensional environment map.

In one embodiment, the present invention provides a method for generating a reflection vector that indexes a three-dimensional environment map. This method comprises the steps of (1) receiving an eye vector and a normal vector neither of which need be normalized, (2) producing a reflection vector without vector normalization, and (3) determining the location where the reflection vector indexes the three-dimensional environment map.

In accordance with another aspect of the present invention, an apparatus is provided for generating a reflection vector that indexes a three-dimensional environment map. The apparatus includes a reflection vector generator for receiving an eye vector and a normal vector neither of which need be normalized. This reflection vector generates a reflection vector without vector normalization. The reflection vector generator then couples to a decoder to supply the generated reflection vector. The decoder, in turn, determines a location where the reflection vector indexes the three-dimensional environment map and in turn, a light shading value is retrieved from the three-dimensional environment map, which is typically stored in random access memory.

Interpolation between map indexes at polygon vertices is unnecessary because the reflection vector can be computed at every pixel since the computationally intensive process of normalization is not necessary at all. Consequently the invention does not produce distorted reflections in computer graphics images and moreover there is no need for polygon subdivision since the choice of the 2-dimensional map selected by a reflection vector is made for each pixel of a polygon. Moreover, the invention allows for a moveable viewer, allowing realistic, fast, real-time interactive computer graphics with rich reflection shadings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 sets forth a vector diagram of an object's diffuse and specular reflection characteristics due to directional point light sources.

FIG. 2 presents a computer system upon which one embodiment of the present invention is implemented.

FIG. 3 presents one embodiment of the reflection vector shading method of the present invention.

FIG. 4 presents another embodiment of the reflection vector shading method of the present invention.

FIG. 5 presents yet another embodiment of the reflection vector shading method of the present invention.

FIG. 6 presents a vector diagram of the generated reflection vector of the present invention indexing a cubic environment map which is aligned with the axis of the coordinate system.

FIG. 7A presents one type of octahedron map whose surface coordinates are described by $|X|+|Y|+|Z|=1$. FIGS. 7B and 7C present the eight faces of the octahedron unfolded into a square to match conventional texture mapping hardware.

FIG. 8 presents still another embodiment of the reflection vector shading method of the present invention.

FIG. 9 presents a vector diagram of the generated reflection vector of the present invention indexing an octahedron environment map which is aligned with the axis of the coordinate system.

FIG. 10 presents a block diagram for one embodiment of a reflection vector generating apparatus of the present invention.

FIG. 11 presents one embodiment of a reflection vector generator used in the reflection vector generating apparatus of FIG. 10.

FIG. 12 presents an embodiment of the decoder used in the reflection vector generating apparatus of FIG. 10.

DETAILED DESCRIPTION OF THE INVENTION

The present invention provides a method and an apparatus for generating reflection vectors which can be unnormalized (i.e., can have non-unit lengths) and for using these reflection vectors to index locations on an environment map. In the following description for purposes of explanation numerous details are set forth in order to provide a thorough understanding of the present invention. However, it will be understood by one of ordinary skill in the art that these specific details are not required in order to practice the invention. In other instances, well-known electrical structures and circuits (such as adders, subtracters, multipliers, etc.) are shown in block diagram form in order not to obscure the description of the present invention with unnecessary detail.

FIG. 2 presents a computer system upon which one embodiment of the present invention is implemented. Computer system **300** includes bus **305** for communicating information. A processor **310** couples with bus **305** for processing digital data. Computer system **300** also includes a random access memory (RAM) **315** coupled to bus **305** for storing digital data and program instructions for execution by processor **310**. Computer system **300** further includes a read only memory (ROM) **320** coupled to bus **305** for storing static information and instructions for processor **310**. In addition, mass data storage device **325**, such as a magnetic disk or an optical disk and its corresponding disk drive, may also be included in the system.

Alphanumeric input device **335** (e.g., a keyboard) may also be coupled to bus **305** for communicating information and command selections to processor **310**. An additional user input device which may be coupled to bus **305** is cursor controller **340**. Input device **340** may take many different forms, such as a mouse, a trackball, a stylus tablet, a touch-sensitive input device (e.g., a touchpad), etc. Another device which may be coupled to bus **305** is hard copy device **345** which may be used for printing a hard copy on paper.

Computer system **300** further includes a display device **330**, such as a cathode ray tube (CRT) or a liquid crystal display (LCD), for displaying information to a computer user. Display device **330** couples to bus **305** via frame buffer **350**, which stores the pixel data for driving the display device **330**. This stored pixel data is generated by graphics subsystem **355**, which is coupled to both processor **310** and frame buffer **350**. It should be noted that alternative embodiments of graphics subsystem **355** embody frame buffer **350**.

One embodiment of graphics subsystem **355** includes a storage means (e.g. a Dynamic Random Access memory (DRAM)) for storing one or more environment maps. Graphics subsystem **355** could further include processing means for performing perspective correct texture mapping (for performing per pixel divisions) during the rendering process. The graphics subsystem **355** would further include processing elements for transforming between coordinate systems and for performing other processing needs of the present invention.

The rendering process of the present invention begins by the processor **310** providing graphics subsystem **355** with the visual description of three-dimensional objects in the scene being rendered. This visual description takes the form of drawing commands and world coordinate vertex data for the geometric entities that form the geometric primitives of the image. The world coordinate system is the coordinate system in which the three-dimensional objects are described. The world coordinate vertex data describes the object's geometric position, color, and surface normal vectors. The graphics subsystem **355** then performs transformations and other graphics operations to calculate specific pixel values for each of the pixels on display device **330**.

In one embodiment of graphics subsystem **355**, the object data from the processor is processed in a four stage pipeline before being displayed on the screen. These four pipeline stages include: 1) a Geometry Subsystem, 2) a Scan Conversion Subsystem, 3) a Raster Subsystem, and 4) a Display Subsystem. The Geometry Subsystem receives the graphical data from the processor to generate screen-space data, which define an object's positions in a screen coordinate system corresponding to the visible plane of the display monitor screen. The Scan Conversion Subsystem then breaks down points, lines, polygons, and meshes to thereby produce pixel data. One embodiment of the present invention is implemented in the Scan Conversion Subsystem. This pixel data is sent to the Raster Subsystem where a z-buffer removes hidden surfaces. The Raster Subsystem also performs various blending and texturing functions on a pixel-by-pixel basis as the pixels are written to the frame buffer. Finally, the Display Subsystem reads the frame buffer and displays the image on a color monitor. Finally, it should be noted that any other configuration for computer system **300** may be used in conjunction with the present invention.

FIG. 3 sets forth one general embodiment of the reflection vector generation method of the present invention. As shown in this figure, at step **405**, the process produces for a point P on a geometric primitive a three-component reflection vector which can be unnormalized. At step **406**, the corresponding 2-dimensional (2-D) map is identified. This corresponding 2-D map is a portion of the environment map and corresponds to the direction of the reflection vector. At step **410**, the three components of the reflection vector are reduced to two components by only preserving the direction information of the reflection vector. Finally, at step **415**, the direction information of the reflection vector is used to ascertain the location in the corresponding 2-D map. This direction information typically provides an address which is used to index the corresponding 2-D map at the indexed location. This indexed location on the 2-D environment map can then be supplied to conventional texture mapping algorithms or devices, which then determine the surface shading attributes for the pixel representing displayed point P on the display device. For example, the texture mapping device could determine the pixel shading attributes by averaging the intensity values within the indexed region of the environment map.

FIG. 4 sets forth one embodiment of the reflection vector generation method of the present invention. As shown in this figure, at step **505**, an eye vector E and a surface normal vector N, for a graphical object's particular displayed point P (i.e., a point on the object that is represented by a pixel on the display device), are received. Both of these vectors can be unnormalized and in the preferred they are both unnormalized in order to avoid the computations involved in obtaining normalized vectors. At step **510**, a reflection vector is produced. This reflection vector is an unnormalized vector when either the normal vector or the eye vector is unnormalized. At step **512**, the corresponding 2-dimensional (2-D) map is identified. This corresponding 2-D map is a portion of the environment map and corresponds to the direction of the reflection vector. At step **515**, the components of the reflection vector are reduced to two components to provide the 2-dimensional index values which are used to locate, at the indexed location, the shading values in the corresponding 2-D map. In step **520**, at least one shading value is obtained from the 2-D map at the indexed location. This shading value at the indexed location on the environment map can then be supplied to conventional texture mapping algorithms or devices, which then determine the surface shading attributes for the pixel representing displayed point P on the display device. For example, the texture mapping device could determine the pixel shading attributes by averaging the intensify values within the indexed region of the environment map.

FIG. 5 sets forth a more detailed embodiment of the present invention's method for generating reflection vectors and for using these reflection vectors to index locations on a cubic environment map which is aligned with the coordinate system which specifies the reflection vector. As shown in this figure, at step **605**, an eye vector E and a surface normal vector N, for a graphical object's particular displayed point P (i.e., a point on the object that is represented by a pixel on the display device), are received. Both of these vectors can be unnormalized. At step **610**, a reflection vector is produced. This reflection vector is an unnormalized vector when either the normal vector or the eye vector is unnormalized. Moreover, this reflection vector is represented by three Cartesian coordinates R_x, R_y, and R_z. In one embodiment of the present invention, the reflection vector is produced by using the following equation:

$$R=2*N*(N\cdot E)-E*(N\cdot N), \quad \text{(iii)}$$

which, as mentioned below, is derived from the prior art equation (ii).

As mentioned before, a cubic environment map can be created by taking six pictures of the environment that surrounds an object with a camera. This cubic environment map is aligned with the world coordinate axes, so that the largest coordinate of a normalized reflection vector indicates the appropriate side to index. Alternatively, a cubic environment map can be generated by the computer (1) by selecting a center of projection, (2) rendering the objects in the computer generated environment for the six cube faces, and (3) recording these images as the cubic environment map. The six faces of a cubic environment map (which are selectively referenced by three-dimensional reflection vectors) can be stored as six two-dimensional maps in conventional two-dimensional texture mapping hardware.

As further shown in FIG. **5**, at step **615**, the 2-D map containing the location corresponding to the direction of the reflection vector is determined. Because the cubic environment map is aligned with the axis of the coordinate system the largest coordinate of the reflection vector and the sign of this coordinate indicate the face of the map that the reflection vector indexes. Consequently, at step **615**, the face of the environment map that the reflection vector indexes is derived by (1) determining the component of the reflection vector (i.e., R_x, R_y, and R_z) that has the largest magnitude, and (2) determining the sign of this largest component. At step **620**, the location where the reflection vector indexes the referenced face of the map is deduced by dividing the two smaller components of the reflection vector by the magnitude of the largest component. This dividing step provides two values which specify the indexed location.

An example of steps **615** and **620** is shown in FIG. **6** (which presents a vector diagram of a generated reflection vector of the present invention indexing a cubic environment map which is aligned with the axis of the coordinate system). The reflection vector is represented by the coordinates 2.5, 1.0, −2.0, in this coordinate system. At step **615** a determination is made that the reflection vector indexes the x equals one face of the cubic environment map by (1) determining that the x coordinate of the reflection vector has the largest magnitude, and (2) determining that the sign for the x coordinate is positive. Thus, the X=+1 face is the indexed face in this case. At step **620**, the R_y and R_z components are divided by the magnitude of the R_x component to determine the indexed location on the x equals one face of the environment map (which in this example is at y=0.4 and z=−0.8). The values for the indexed location on the indexed face are then used in the conventional manner to retrieve the appropriate shading values from the indexed face of the map.

The indexed location on the selected 2-D map can then be supplied to conventional texture mapping algorithms or devices, which then determine the surface shading attributes for the pixel representing displayed point P on the display device. For example, the texture mapping device could determine the pixel shading attributes by averaging the intensity values within the indexed region of the indexed face of the 2-D map.

Cubic environment maps are not the only type of environment maps that can be used in the present invention. In fact, a three dimensional map consisting of any set of 2-D maps which cumulatively includes one and only one location for every possible direction from the center of the environment could be used in the present invention, because such a set of maps allows a unique location on it to be uniquely identified by a the direction of the reflection vector. For example, as further discussed below by reference to FIG. **8**, octahedron environment maps can be utilized in the present invention. The use of 2-D maps is advantageous, because two-dimensional images require far less storage than three-dimensional volume tables.

FIG. **7A** sets forth one type of octahedron map. This octahedron map has vertices which lie on the Cartesian axes and has its surface coordinates described by $|X|+|Y|+|Z|=1$. This map further has eight faces which differ from each other in the signs of their x, y, and z coordinates. For example, all the points with positive x and negative y and z which solve the above equation form one face. So given the sign bits of the three components of the reflection vector R_x, R_y, and R_z, a particular face of the octahedron can be selected. Furthermore, if viewed orthographically along one of the coordinate axis, for example from +Y or from −Y, the four visible faces in each case look like 45-degree right triangles, which can be indexed by their surface X and Z positions alone. Moreover, as shown in FIGS. **7B** and **7C**, the eight faces of the octahedron can be unfolded into a

square to match conventional texture mapping hardware, because these faces form 45-degree right triangles when they are viewed orthographically. Thus, as set forth below in Table 1, the eight faces of the octahedron map to triangles in a square environment texture map which has indexes s and t.

conventional two-dimensional mapping storage (as shown in FIGS. 7B and 7C), the location where the reflection vector indexes the referenced face of the map is deduced by dividing the R_x and R_z components of the reflection vector by the sum of the magnitudes of the reflection vector components ($|R_X|+|R_Y|+|R_Z|$).

TABLE 1

Face	X -to- S mapping	C -to- T mapping	Z -to- T mapping	Z -to- S mapping
1 −X,+Y,+Z	X = −1.0 . . . 0.0		Z = −0.0 . . . 1.0	
	S = 0.0 . . . 0.5		T = 0.5 . . . 1.0	
2 +X,+Y,+Z	X = 0.0 . . . 1.0		Z = 0.0 . . . 1.0	
	S = 0.5 . . . 1.0		T = 0.5 . . . 1.0	
3 −X,+Y,−Z	X = −1.0 . . . 0.0		Z = −1.0 . . . 0.0	
	S = 0.0 . . . 0.5		T = 0.0 . . . 0.5	
4 +X,+Y,−Z	X = 0.0 . . . 1.0		Z = −1.0 . . . 0.0	
	S = 0.5 . . . 1.0		T = 0.0 . . . 0.5	
5 −X,−Y,+Z		X = −1.0 . . . 0.0		Z = 0.0 . . . 1.0
		T = 0.5 . . . 1.0		S = 0.0 . . . 0.5
6 +X,−Y,+Z		X = −0.0 . . . 1.0		Z = 0.0 . . . 1.0
		T = 1.0 . . . 0.5		S = 0.0 . . . 0.5
7 −X,−Y,−Z		X = 1.0 . . . 0.0		Z = −1.0 . . . 0.0
		T = 0.5 . . . 0.0		S = 0.5 . . . 0.0
8 +X,−Y,−Z		X = 0.0 . . . 1.0		Z = −1.0 . . . 0.0
		T = 0.0 . . . 0.5		S = 0.5 . . . 1.0

FIG. 8 sets forth another embodiment of the present invention's method for generating reflection vectors and for using these reflection vectors to index locations on an octahedron environment map which is aligned with the axes of the coordinate system which also determines the coordinate of the reflection vector. As shown in this figure, at step **1005**, an eye vector E and a surface normal vector N, for a particular displayed point P of a graphical object (i.e., a point on the object that is represented by a pixel on the display device), are received. Both of these vectors can be unnormalized. At step **1010**, a reflection vector is produced. This reflection vector is an unnormalized vector when either the normal vector or the eye vector is unnormalized. Moreover, this reflection vector is represented by three Cartesian coordinates R_x, R_y, and R_z. In one embodiment of the present invention, the reflection vector is produced by using the following equation:

$$R=2*N*(N\cdot E)-E*(N\cdot N), \qquad \text{(iii)}$$

which, as mentioned below, is derived from the prior art equation (ii).

As further shown in FIG. **8**, at step **1015** and **1020**, the location where this generated reflection vector indexes (i.e., where this reflection vector or its projection intersects) an octahedron environment map, which is aligned with the axes of the coordinate system, is determined. As mentioned before, the eight faces of an octahedron environment map differ from each other in the signs of their x, y, and z coordinates. Consequently, at step **1015**, the face of the environment map that the reflection vector indexes is derived by examining the sign bits of the three components of the reflection vector R_x, R_y, and R_z.

Furthermore, unlike cubic environment map projection which involves dividing the largest magnitude component into the other two, octahedron projection involves dividing the sum of the magnitudes of the components ($|R_X|+|R_Y|+|R_Z|$) into two of the components. Thus, for the embodiments of the present invention which index an octahedron environment map by using x and z coordinates to reference a

An example of steps **1015** and **1020** is shown in FIG. **9** (which presents a vector diagram of a generated reflection vector of the present invention indexing an octahedron environment map which is aligned with the axis of the coordinate system). The reflection vector is represented by the coordinates 2.5, 1.0, −2.0. At step **1015** a determination is made that the reflection vector indexes face four of the octahedron environment map by examining the sign bits of the reflection vector components. At step **1020**, the R_x and R_z components are divided by the magnitude of the sum of the reflection vector components (i.e., divided by 5.5) to determine the indexed location on face four of the environment map (which in this example is at x=0.455 and z=−0.364). Finally, at step **1025**, the x and z coordinates are transformed into s and t coordinates to obtain the shading values for the referenced location of the octahedron environment map from the two-dimensional texture mapping storage.

FIG. **10** sets forth one embodiment of the reflection vector shading hardware of the present invention. As shown in this figure, reflection vector shading hardware **1200** includes reflection vector generator **1205** and decoder **1210**. In order to generate a reflection vector, reflection vector shading hardware **1200** receives an eye vector and a normal vector for an eye vector E and a surface normal vector N, for a particular displayed point P of a graphical object (i.e., a point on the object that is represented by a pixel on the display device). This reflection vector is unnormalized when either the eye vector or the normal vector is unnormalized. As mentioned before, if the normal and the eye vectors are normalized, the reflection vector is represented by the following prior art equation:

$$R_n=2*N_n*(N_n\cdot E_n)-E_n. \qquad \text{(ii)}$$

Since a vector is normalized by dividing the vector by its length, this prior art equation (ii) can be replaced by the following equation:

$$Rn = 2*\frac{N}{|N|}*\frac{(N \cdot E)}{|N||E|} - \frac{E}{|E|}.$$

This transformation results in the multiplication of two |N|'s in the denominator, which equals (N·N). Substituting (N·N) for |N|*|N| yields:

$$Rn = 2*\frac{N*(N \cdot E)}{(N \cdot N)*|E|} - \frac{E}{|E|}.$$

But since the reflection vector R does not need to be normalized, both sides of the above equation can be multiplied by |E|*(N·N), which results in the following equation:

$$\begin{aligned} R &= R_n*(N \cdot N)*|E| \\ &= 2*N*(N \cdot E) - E*(N \cdot N). \end{aligned} \quad \text{(iii)}$$

Consequently, in one embodiment of the present invention, reflection vector generator **1205** uses the relationship set forth in equation (iii) to generate the reflection vector. By using equation (iii), reflection vector generator **1205** can generate an unnormalized reflection vector by only using two dot products, two vector scaling operations, and a vector subtraction.

FIG. 11 sets forth one embodiment of reflection vector generator **1205** of FIG. **10**. As shown in FIG. **11**, reflection vector generator **1300** can be implemented with a number of multipliers **1305**, **1310**, **1315**, **1320**, and **1325**, a number of adders **1330** and **1335**, and a subtracter **1340**. First set of multipliers **1305** and adder **1330** generate the (N·E) component of equation (iii), while second set of multipliers **1310** and adder **1335** generate the (N·N) component of equation (iii). Third set of multipliers **1315** and times two multiplier **1325** then generate the 2*N*(N·E) component of equation (iii), while the fourth set of multipliers **1320** generates the E*(N·N) component of equation (iii). Finally, subtracter **1340** subtracts the output of the fourth set of multiplier **1320** from the output of times two multiplier **1325** in order to produce the above-mentioned equation (iii).

Thus, in one embodiment of the present invention, reflection vector generator **1205** of FIG. **10** uses the above-mentioned equation (iii) to generate a reflection vector based on the normal and the eye vectors which need not be normalized. Reflection vector generator **1205** then supplies the generated reflection vector to decoder **1210**. In turn, decoder **1210** determines the location where the reflection vector indexes a three-dimensional environment map in the manner described above (e.g. steps **615** and **620** of FIG. 5). Decoder **1210** of FIG. **10** then supplies this indexed location to a conventional texture mapping device, which in turn projects the two-dimensional image indexed on the environment map onto particular pixel P.

For the embodiments of reflection vector shading hardware **1200** that are designed to operate with a cubic environment map which is aligned with the axis of the coordinate system which specifies the reflection vector, FIG. **12** sets forth an embodiment of decoder **1210**. As shown in this figure, decoder **1500** includes select logic **1510**, multiplexor **1515**, divider **1520**, and two multipliers **1525**. This decoder receives the x, y, and z components of the generated reflection vector from reflection vector generator **1205**, in order (1) to determine the indexed face of a cubic environment map that the reflection vector selects/indexes, and (2) to determine the indexed location on this indexed face.

The select logic **1510** is coupled to receive the components Rx, Ry, Rz of the reflection vector and determines the component of the reflection vector that has the largest magnitude. The select logic unit **1510** also determines the sign of the reflection vector's component with the largest magnitude. Based on these determinations, select logic **1510** (1) supplies a two-bit signal to multiplexor **1515** in order to cause this multiplexor to identify the axis of this largest coordinate as the major axis, and (2) supplies a three-bit signal on line **1530**, which identifies the indexed face of the cubic environment map, to conventional texture mapping hardware.

Multiplexor **1515** then supplies the major axis coordinate to divider **1520** which produces the reciprocal of the inputted major coordinate value. Based on a pre-determined relationship that is pre-programmed in the multiplexor, multiplexor **1515** then defines the axis of the two smaller coordinates as a first minor axis and a second minor axis. Multiplexor **1515** then supplies these minor axes' coordinates to multipliers **1525**. These multipliers in turn generate the indexed location on the indexed face of the cubic environment map by multiplying the two minor coordinates by the output of divider **1520**.

It will be appreciated that the apparatus of the invention may be a general purpose processor (e.g. a microprocessor or a microcontroller or an ASIC) which is controlled by software to generate the unnormalized reflection vector at a pixel location and then to decode this vector to produce a location indexed by this vector in a selected 2-dimensional map in the environment map. This embodiment may be implemented by having the software instruct the processor to compute the reflection vector in the manner described above (e.g. equation iii) and to decode the vector also in the manner described above.

As mentioned before, computer graphic systems have long sought to incorporate surface reflections in their computer graphic models, because surface reflections are useful for appraising the smoothness and the curvature of complex surfaces. By utilizing the above-described teachings of the present invention, computer graphic systems can now generate surface reflections at real time speeds allowing the user to interactively manipulate the computer images presented by such systems. More specifically, a computer graphic system can now allow its viewers to inspect the surface curvature and smoothness of its model (1) by generating reflection vectors, which can be unnormalized, for every displayed point of its model, and (2) by shading its model with the images that these reflection vectors index on an environment map.

Alternatively, if a computer generated object is relatively far from the viewer (i.e., if the eye vector changes relatively slowly across the surface of the computer generated object), a computer graphic system can render this object with rich surface reflections by (1) generating reflection vectors, that can be unnormalized, for the vertices of the polygons representing this object, (2) generating the remaining reflection vectors for this object through linear interpolation, and (3) shading this object with the images that these generated reflection vectors index on an environment map.

The present invention enables computer graphic systems to produce rich surface reflections at real time speeds (i.e., at speeds that allow the viewer to move interactively with respect to the computer generated object), because the present invention provides an environment mapping method which does not require the normalization of the reflection vectors. In fact, if computer graphic systems employ the embodiments of the present invention that utilize the above-mentioned equation (iii) to generate reflection vectors, these systems can render rich surface reflections at real time speeds by only computing two dot products, two vector scaling products, and a vector subtraction for each reflection vector.

The present invention is further advantageous because it provides a method and an apparatus for rendering high-quality specular highlights in real time and independent of the number of light sources. As mentioned above, because the present invention provides a shading method which does not require the normalization of the reflection vectors, the present invention enables computer graphic systems to produce rich surface reflections (such as specular highlights) at real time speeds allowing interactive use by a user of the computer system.

In addition, since environment maps are pre-computed and since these maps can contain any number of area light sources, the present invention does not suffer from any rendering-time penalty for rendering specular highlights when there is a high number of light sources. Thus, by incorporating an arbitrary number of light sources into an environment map that is indexed by reflection vectors that are generated at real time speeds, the present invention can render high quality specular highlights in real time and independent of the number of light sources. Specular spread functions can be incorporated into the environment map itself if they are based solely on relative reflection angle. Thus, a Phong highlight can be incorporated, allowing real-time interactive Phong shading.

Furthermore, the embodiments of the present invention that generate index values for a cubic environment map are also advantageous because these embodiments do not need to utilize perspective-correcting texture hardware. When a computer graphic system incorporates perspective viewing in its model the reflection direction has to be projected from screen space back to the world space of the environment map, in order to insure that the reflection vector correctly indexes the environment map. Perspective viewing is accomplished by using homogenous factor 1/w (where w is proportional to the distance between a viewer and a displayed point) to substitute N/w and E/w for the normal and the eye vectors iterated in screen space. These two substituted vectors are then used to calculate R/w.

Prior art environment mapping techniques then employ perspective-correct texture divider to divide the result of this calculation (i.e., R/w) by 1/w at each pixel in order to return the reflection vector to the world space coordinates direction. However, the embodiments of the present invention that generate index values obviate the need for this divider, because in these embodiments the w's in the denominators of each component of R/w cancel. These embodiments of the present invention then can use the prior art perspective-correct divider for index calculations.

One of ordinary skill in the art would recognize that the above-described invention may be embodied in other specific forms without departing from the spirit or the essential characteristics of the disclosure. For instance, in alternative embodiments of the present invention, other polygonal-faced surfaces, such as dodecahedron or isocahedron can be used as environment maps. Thus, while certain exemplary embodiments have been described and shown in the accompanying drawings, the invention is not to be limited by the foregoing illustrative details but rather is to be defined by the appended claims.

What is claimed is:

1. A method for generating a reflection vector, said method comprising the steps of:

a) receiving an eye vector (E) and a normal vector (N) at least one of which is not normalized; and

b) producing a reflection vector (R) without vector normalization.

2. The method of claim 1 further comprising the step of determining the location where the reflection vector indexes an environment map.

3. The method of claim 2 wherein said method is used to render specular highlights.

4. The method of claim 1, wherein R=2*N*(N·E)–E*(N·N).

5. A method for generating a reflection vector that indexes one of a plurality of faces of a cubic environment map stored in a memory, said method comprising the steps of:

receiving an eye vector (E) and a normal vector (N);

producing a reflection vector (R) represented by a first coordinate, a second coordinate, and a third coordinate, wherein the reflection vector is unnormalized when one of the normal vector and the eye vector is unnormalized;

determining a largest coordinate having the largest magnitude of said first, second, and third coordinates; and

generating a fourth coordinate, and a fifth coordinate, by dividing two of said first, second, and third coordinates by said largest coordinate, wherein said fourth, and fifth coordinates define the location on the cubic environment map where said reflection vector indexes said cubic environment map in said memory.

6. The method for generating a reflection vector of claim 5, wherein R=2*N*(N·E)–E*(N·N).

7. In a data processing system which renders images, a method of generating reflections off a rendered object by using an environment representation comprising a set of 2-dimensional maps which are independent of a viewer's position, said method comprising:

receiving an eye vector (E) and a normal vector (N), at least one of which is not normalized;

producing a reflection vector (R) without vector normalization;

selecting a selected 2-dimensional map which contains a location indexed by said reflection vector;

determining said location on said selected 2-dimensional map; and

sampling a reflection shading value at said location in said selected 2-dimensional map.

8. The method of claim 7, wherein R=2*N*(N·E)–E*(N·N).

9. A method as in claim 7 wherein said environment representation comprises six 2-dimensional maps arranged as the faces of a cube which is aligned to an axis of a coordinate system, and wherein said step of selecting comprises selecting the 2-dimensional map corresponding to the largest component of said reflection vector and wherein said step of determining comprises dividing the two smaller components of said reflection vector by said largest component.

10. A method as in claim 7 wherein said environment representation comprises eight 2-dimensional maps arranged as the triangular faces of an octahedron which is aligned to an axis of a coordinate system, and wherein said step of selecting comprises selecting the 2-dimensional map corresponding to the direction indicated by the signs of a plurality of components of said reflection vectors and wherein said step of determining comprises dividing a plurality of magnitudes of said plurality of components by a sum of the magnitudes of said plurality of components.

11. A reflection vector generating apparatus for generating a reflection vector indexing one of a plurality of faces of an environment map, said apparatus comprising:

a) a reflection vector generator which receives an eye vector and a normal vector at least one of which is not normalized, said reflection vector generator producing a reflection vector without vector normalization;

b) a decoder coupled to the reflection vector generator to receive the reflection vector, said decoder determining the face of said environment map that said reflection vector indexes.

12. An apparatus as in claim 11 wherein said reflection vector generator comprises a plurality of multipliers and a shifter.

13. An apparatus as in claim 11 wherein said decoder comprises a multiplexer and a divider.

14. An apparatus as in claim 11 wherein said reflection vector generator and said decoder comprise one of a microprocessor or a microcontroller which is controlled by software to generate said reflection vector and to determine the face of said environment map which said reflection vector indexes.

15. An apparatus as in claim 11 wherein said decoder further determines an indexed location on said face.

16. An apparatus as in claim 11 further comprising:

memory coupled to said decoder, said memory storing said environment map;

a display, said display displaying a rendered object having reflections which are produced by said reflection generator;

a processor coupled to said display, said processor calculating said eye vector and said normal vector and providing said eye and normal vectors to said reflection vector generator.

17. A method as in claim 7, wherein said rendered object includes a first pixel, and wherein said method further comprises calculating a pixel shading value for said first pixel using said reflection shading value and displaying on said display device said first pixel having said pixel shading value.

18. A method as in claim 7 wherein said method is used to render specular highlights.

19. A method as in claim 18 wherein said environment representation includes a representation of specular highlights.

* * * * *

APPENDIX D

Cerny (US Patent Pub. 2003/0112238), cited by the Examiner in the Final Office Action dated June 8, 2009.

US 20030112238A1

(19) United States
(12) Patent Application Publication
(10) Pub. No.: US 2003/0112238 A1
(43) Pub. Date: Jun. 19, 2003

Cerny et al.

(54) SYSTEM AND METHOD FOR ENVIRONMENT MAPPING

(76) Inventors: Mark Evan Cerny, Los Angeles, CA (US); Pal-Kristian Engstad, Los Angeles, CA (US)

Correspondence Address:
CARR & FERRELL LLP
2225 EAST BAYSHORE ROAD
SUITE 200
PALO ALTO, CA 94303 (US)

(21) Appl. No.: 10/267,341

(22) Filed: Oct. 8, 2002

Related U.S. Application Data

(60) Provisional application No. 60/328,490, filed on Oct. 10, 2001.

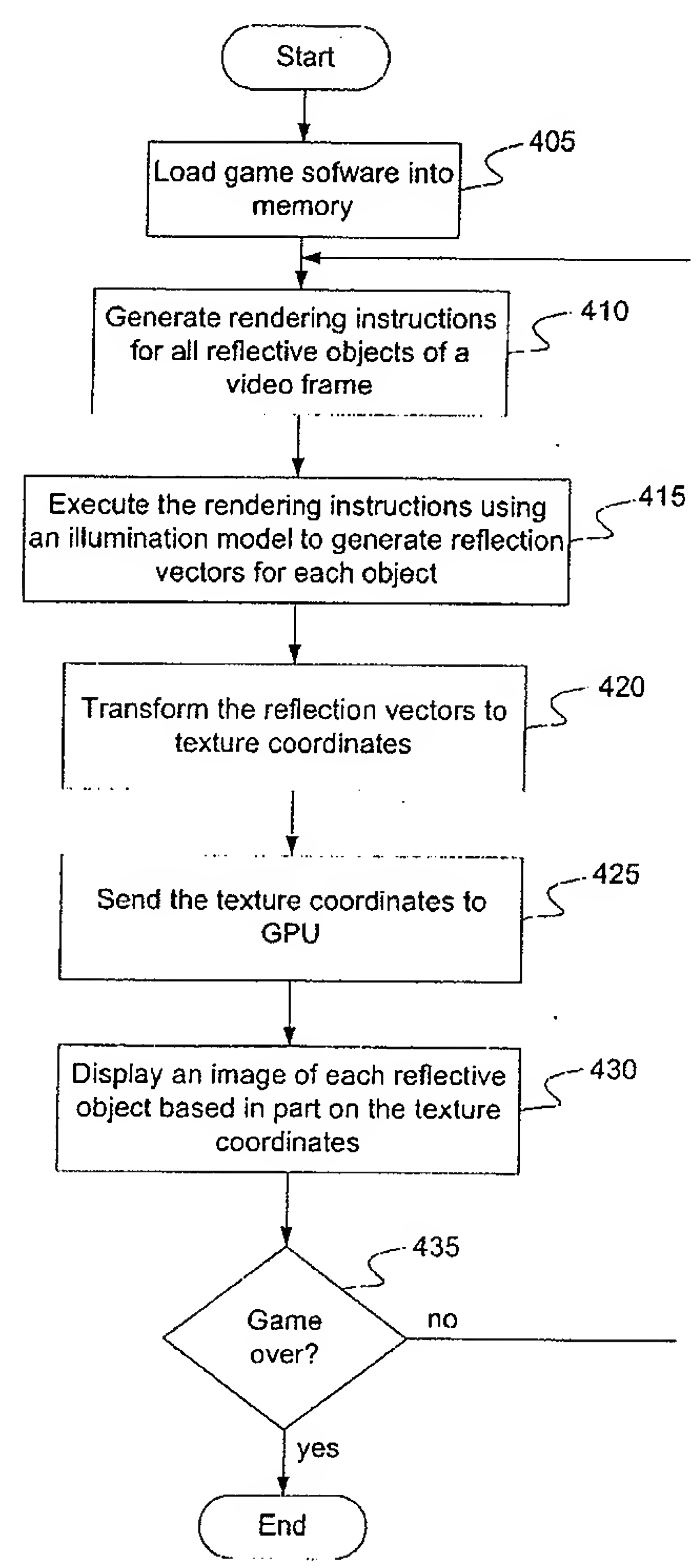
Publication Classification

(51) Int. Cl.[7] G06T 15/60
(52) U.S. Cl. 345/426

(57) ABSTRACT

A system and method for environment mapping determines a computer-generated object's reflective appearance, based upon position and orientation of a camera with respect to the object's location. The present invention is implemented as a real-time environment mapping for polygon rendering, however, the scope of the invention covers other rendering schemes. According to one embodiment of the present invention, a vector processing unit (VPU) uses a modified reflection formula to compute reflective properties of an object. The modified reflection formula is:

$r=e-(e\cdot(n+e_o))(n+e_o)/(1-nz)=e-(e\cdot[nx, ny, nz-1])[nx, ny, nz-1]/(1-nz)$, where $e_o=[0,0,-1]$, and nx, ny, and nz are the components of the surface normal vector, n.



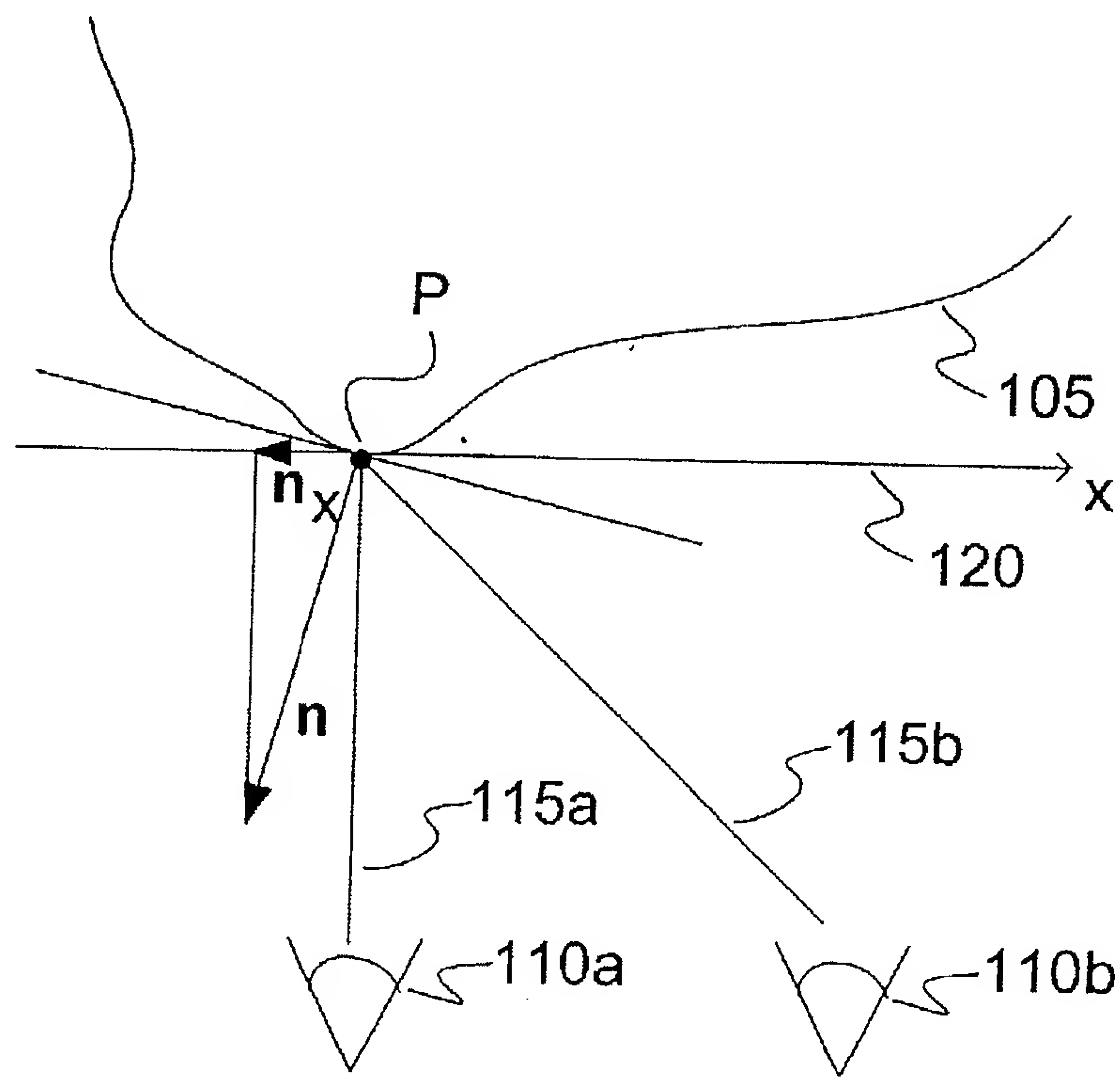


FIG. 1

Prior Art

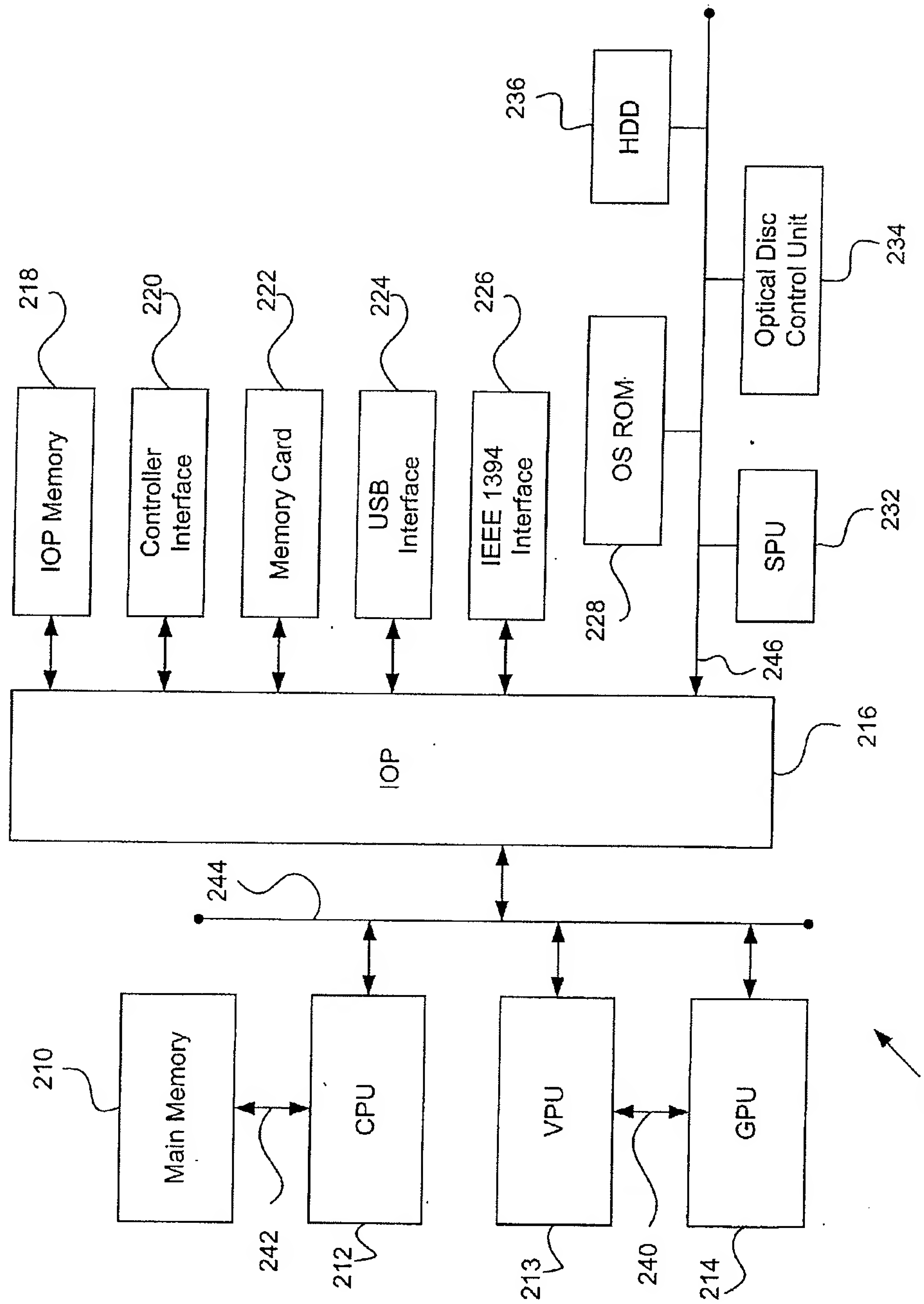

HDD
236
Optical Disc
Control Unit
234
OS ROM
228
SPU
232
246
IOP Memory
218
Controller
Interface
220
Memory Card
222
USB
Interface
224
IEEE 1394
Interface
226
IOP
216
244
Main Memory
210
242
CPU
212
VPU
213
240
GPU
214
200

FIG. 2

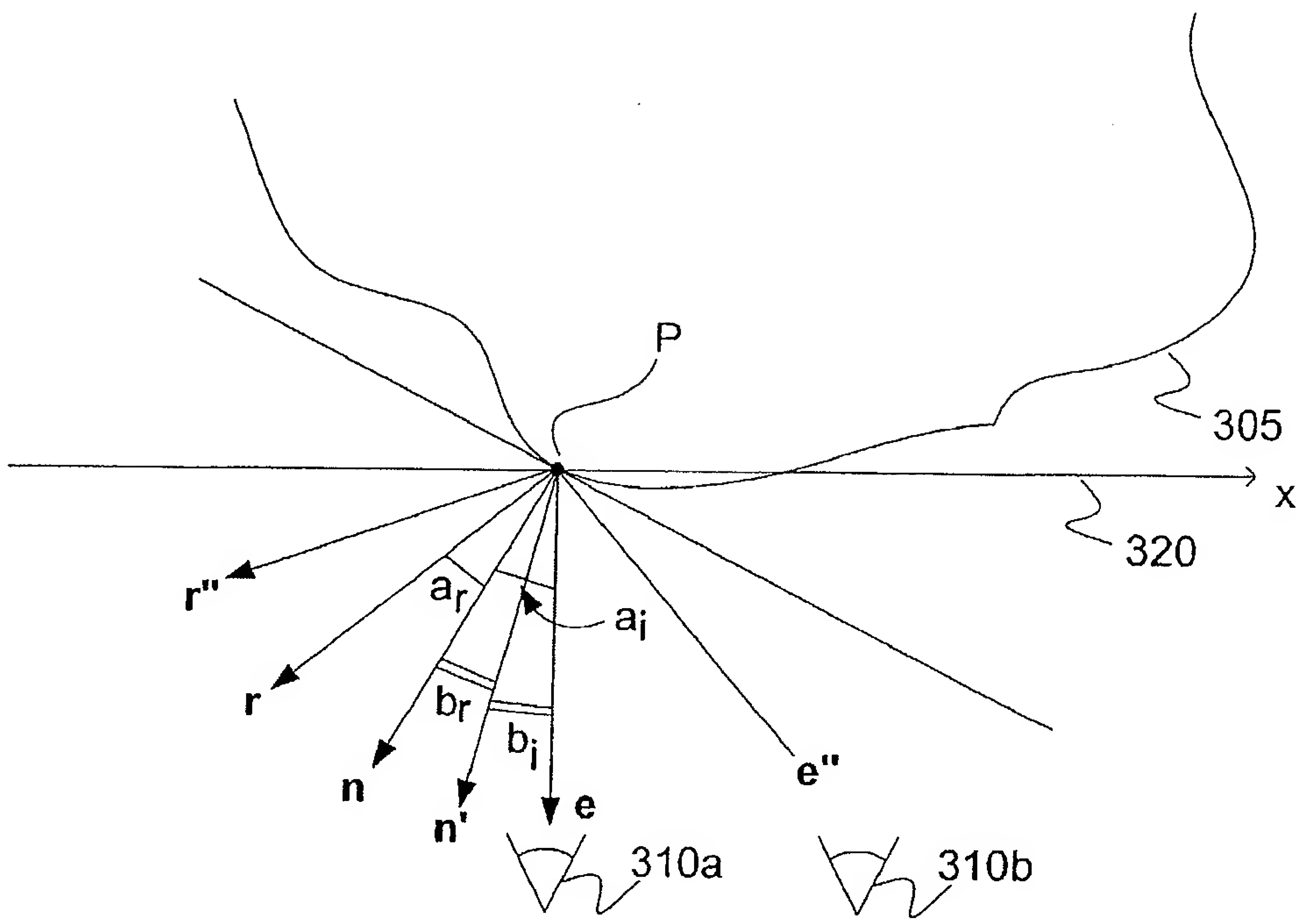


FIG. 3

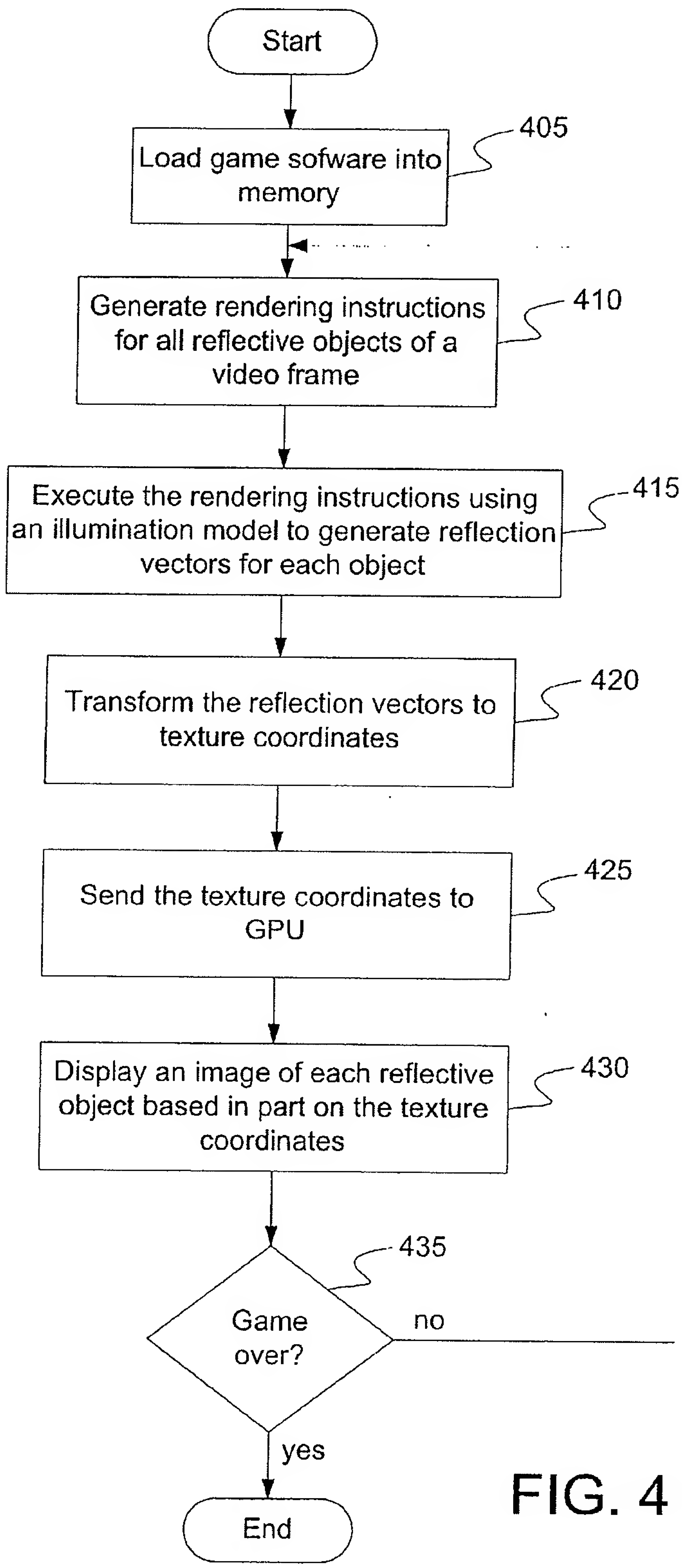
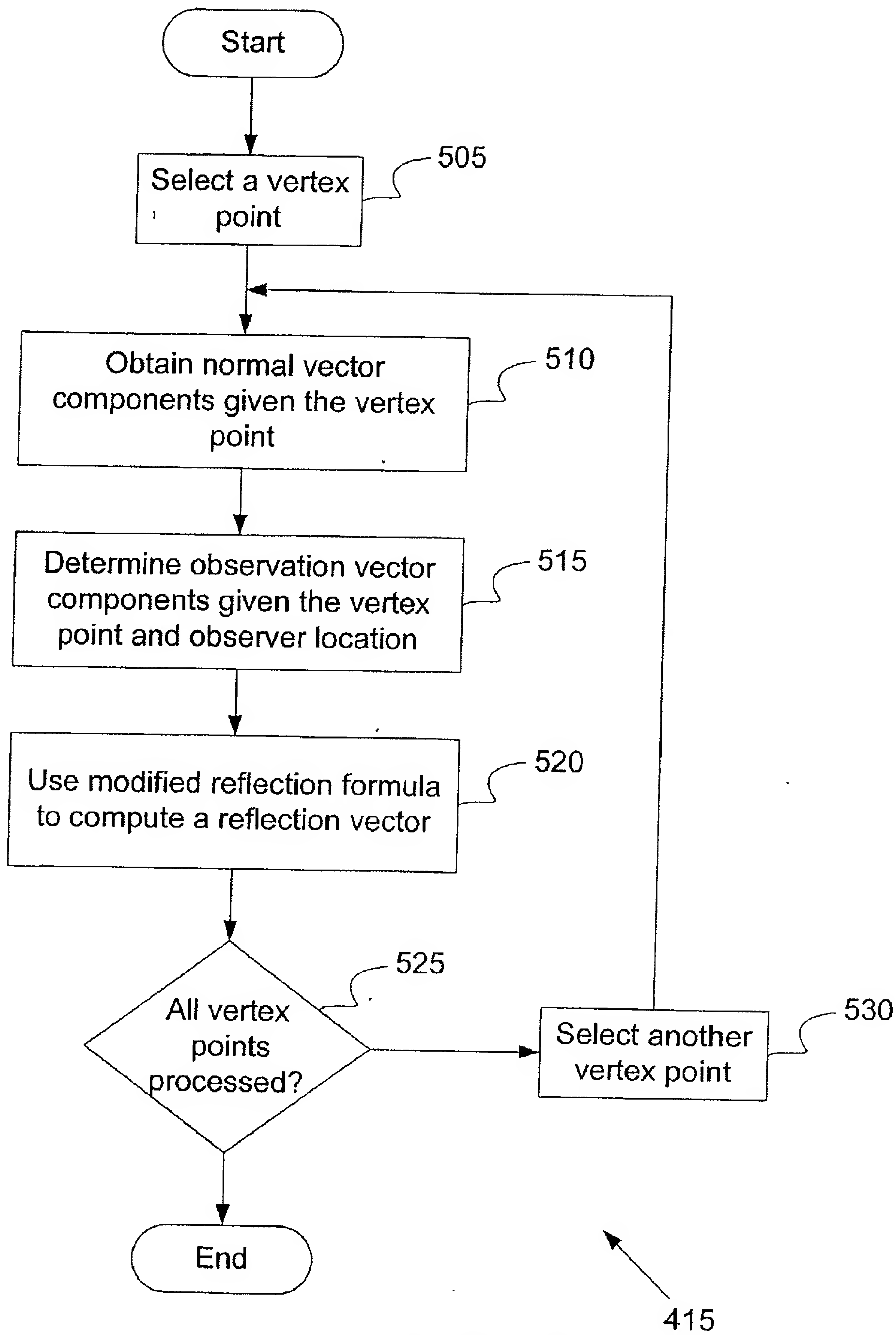

Start
Load game sofware into memory
405
Generate rendering instructions for all reflective objects of a video frame
410
Execute the rendering instructions using an illumination model to generate reflection vectors for each object
415
Transform the reflection vectors to texture coordinates
420
Send the texture coordinates to GPU
425
Display an image of each reflective object based in part on the texture coordinates
430
Game over?
435
no
yes
End

FIG. 4



Start
Select a vertex point
505
Obtain normal vector components given the vertex point
510
Determine observation vector components given the vertex point and observer location
515
Use modified reflection formula to compute a reflection vector
520
All vertex points processed?
525
Select another vertex point
530
End
415

FIG. 5

SYSTEM AND METHOD FOR ENVIRONMENT MAPPING

CROSS REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of U.S. Provisional Patent Application Serial No. 60/328,490, entitled "Environment Mapping," filed on Oct. 10, 2001, which is incorporated herein by reference.

BACKGROUND OF THE INVENTION

[0002] 1. Field of the Invention

[0003] This invention relates generally to computer generated images and more particularly to a system and method for environment mapping.

[0004] 2. Description of the Background Art

[0005] Typically, the illumination of a computer-generated object by discrete light sources, continuous light sources, and ambient light is described by an illumination model. The object is illuminated by the reflection of ambient light and the reflection of light source light from the surface of the object. Generally, the illumination model is a mathematical expression that operates on a set of variables to generate reflection properties, such as color and intensity of reflected light and an object's texture as viewed by an observer. Given ambient light and light sources positioned about the object, the illumination model defines the reflection properties of the object. The illumination model is considered to be accurate if the illuminated object appears realistic to an observer.

[0006] Typically, the illumination model is incorporated in a software program executed by a vector processing unit, a central processing unit, or a rendering engine of a computer system. The program must be capable of computing the illumination of the object when the light sources change position with respect to the object, when the observer views the illuminated object from a different angle, or when the object is rotated. Furthermore, an efficient illumination model is needed for the processing unit to compute the illumination in real-time, for example, if the observer (i.e., a camera) is moving with respect to the object. Therefore, it is desired to incorporate terms in the illumination model that are computationally cost effective, while at the same time generating an image of the illuminated object that is aesthetically pleasing to the observer.

[0007] Computing texture (i.e., environment mapping) is important when rendering a realistic image of the illuminated object that closely resembles a real physical object. Typically, texture coordinates for each point of the object's surface are computed, and a texture map comprising the texture coordinates is generated.

[0008] **FIG. 1** illustrates a prior art direct normal projection method for computing an object's texture coordinates. **FIG. 1** includes an object's surface **105**, a point P on surface **105**, a normal vector n to surface **105** at point P, an observer **110***a*, a line-of sight **115***a* between observer **110***a* and the point P, and a projection of the normal vector n onto an x-axis **120**, referred to as n_x. In general, a z-axis (not shown) is perpendicular to x-axis **120** and is in the plane of **FIG. 1**, and a y-axis (not shown) is perpendicular to x-axis **120** and the z-axis and is out of the plane of **FIG. 1**. For simplicity of illustration, the **FIG. 1** embodiment of object's surface **105** is a line, however, surface **105** is typically any 2-D surface, and hence in general, the normal vector n may have a vector component n_y along the y-axis.

[0009] In operation, the direct normal projection method computes the projected components n_x and n_y of the normal vector n for each point P on object's surface **105**. The central processing unit or vector processing unit then maps (i.e., transforms) the projected components n_x and n_y into texture coordinates (s,t) using one or more mapping algorithms known in the art. The vector processing unit then uses the computed texture coordinates (s,t) for each point P, as well as other reflection variables, in an illumination model to generate a reflection pattern of object's surface **105**. Although the direct normal projection method of the prior art may be fast, the method generates a reflection pattern that appears "painted-on" as observer **110***a* moves to different locations. In other words, the reflection pattern of object's surface **105** does not change with respect to rotation or translation of observer **110***a*, since the method depends upon the x and y components of the normal vector n, independent of the position of observer **110***a* with respect to the point P. For example, the vector processing unit computes the same projected components (n_x,n_y) and texture coordinates (s,t) for an observer **110***b* viewing point P as observer **110***a* viewing point P.

[0010] It would be useful to implement a system and method of environment mapping that depends upon an observer's location with respect to an object's location and orientation to generate a more realistic reflection pattern, and that is consistent with results of the direct normal projection method for particular object-observer geometries.

SUMMARY OF THE INVENTION

[0011] In accordance with the present invention, a system and method for environment mapping of a reflective object is disclosed. In one embodiment of the invention, the method includes constructing a surface normal vector n at a point P on a surface of the reflective object, constructing an observation vector e from the point P to an observer, and using a modified reflection formula to compute a reflection vector r based on the surface normal vector n and the observation vector e. The modified reflection formula is based on reflection about a pseudo-normal vector n' at the point P on the surface.

[0012] According to the present invention, the pseudo-normal vector n' bisects an angle subtended by the surface normal vector n and a reference observation vector e_o, where the reference observation vector e_o is directed from the point P to an observer located directly in front of the point P.

[0013] The modified reflection formula is:

[0014] $r=e-(e\cdot(n+e_o))(n+e_o)/(1-nz)=e-(e\cdot[nx, ny, nz-1])[nx, ny, nz-1]/(1-nz)$, where $e_o=[0,0,-1]$, and nx, ny, and nz are the components of the surface normal vector n. Each computed reflection vector r may be processed to generate a pair of texture coordinates (s,t). The reflective object is then rendered based in part on the texture coordinates (s,t) associated with each point P on the surface of the reflective object. The scope of the present invention

covers all types of rendering schemes, such as a polygon rendering where each point P on the surface of the reflective object is located at the vertex of a polygon.

[0015] In another embodiment of the invention, the system includes a memory configured to store a modified reflection model, a vector processing unit configured to compute reflection vectors using the modified reflection model, and a graphics processor configured to render the reflective object in an image. The quality of the image is dependent upon the texture coordinates that are derived from the computed reflection vectors.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] **FIG. 1** illustrates a prior art direct normal projection method for computing an object's texture coordinates;

[0017] **FIG. 2** is a block diagram of one embodiment of an electronic entertainment system according to the invention;

[0018] **FIG. 3** illustrates a modified reflection projection method stored in main memory **210** of **FIG. 2**, according to one embodiment of the invention;

[0019] **FIG. 4** is a flowchart of method steps for displaying an image of a reflective object based upon texture coordinates, according to one embodiment of the invention; and

[0020] **FIG. 5** is a flowchart of method steps for step **415** of **FIG. 4** to generate reflection vectors for a reflective object, according to one embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0021] The system and method for environment mapping described herein allow a computer-generated object's reflective appearance to change, based upon position and orientation of a camera with respect to the object's location. A position of the camera may be defined by a lateral location of the camera with respect to the object's location. Lateral camera movement is defined as motion to the right, left, up, or down with respect to the object's location. Camera orientation may be defined by rotation angles with respect to a given, fixed coordinate system.

[0022] An exemplary embodiment of the invention is implemented as a real-time environment mapping for polygon rendering. However, the scope of the invention covers other applications, such as environment mapping for other rendering schemes. Other rendering schemes may include, but are not limited to, point-based and non-polygon volume-based primitives. Various embodiments of the invention may be enabled in software, hardware, or firmware.

[0023] According to one embodiment of the invention, a central processing unit (CPU) and/or one or more vector processing units (VPUs) use illumination models to compute reflective properties of an object. The object's reflective properties are associated with the objects' appearance. Reflective properties include color and intensity of light reflected by the object, and texture of the reflective object. The texture of an object is associated with reflective properties such as the object's shininess and overall surface appearance. Typically, the object's texture is specified by texture coordinates (s,t) computed by the VPU. Texture coordinates may be incorporated into a texture map which is wrapped (i.e., mapped) around the object. For example, a VPU may execute environment mapping instructions that operate on variables stored in a VPU random access memory (RAM) or on variables stored in a CPU register to compute the texture coordinates. Typically the texture coordinates and the other computed reflective properties (also referred to as illumination terms) such as color and intensity are passed to a graphics processing unit (GPU) for further processing. Subsequently, the GPU prepares the reflective object for display on a display device such as a computer monitor.

[0024] **FIG. 2** is a block diagram of one embodiment of an electronic entertainment system **200** according to the invention. System **200** includes, but is not limited to, a main memory **210**, a CPU **212**, a VPU **213**, a GPU **214**, an input/output processor (IOP) **216**, an IOP memory **218**, a controller interface **220**, a memory card **222**, a Universal Serial Bus (USB) interface **224**, and an IEEE 1394 interface **226**. System **200** also includes an operating system read-only memory (OS ROM) **228**, a sound processing unit (SPU) **232**, an optical disc control unit **234**, and a hard disc drive (HDD) **236**, which are connected via a bus **246** to IOP **216**.

[0025] CPU **212**, VPU **213**, GPU **214**, and IOP **216** communicate via a system bus **244**. CPU **212** communicates with main memory **210** via a dedicated bus **242**. VPU **213** and GPU **214** may also communicate via a dedicated bus **240**.

[0026] CPU **212** executes programs stored in OS ROM **228** and main memory **210**. Main memory **210** may contain pre-stored programs and may also contain programs transferred via IOP **216** from a CD-ROM or DVD-ROM (not shown) using optical disc control unit **234**. IOP **216** controls data exchanges between CPU **212**, VPU **213**, GPU **214** and other devices of system **200**, such as controller interface **220**.

[0027] Main memory **210** includes, but is not limited to, a program having game instructions including an illumination model. The program is preferably loaded from a DVD-ROM via optical disc control unit **234** into main memory **210**. CPU **212**, in conjunction with VPU **213**, GPU **214**, and SPU **232**, executes game instructions and generates rendering instructions using inputs received from a user via controller interface **220**. The user may also instruct CPU **212** to store certain game information on memory card **222**. Other devices may be connected to system **200** via USB interface **224** and IEEE 1394 interface **226**.

[0028] In one embodiment of the invention, VPU **213** executes instructions from CPU **212** to generate texture coordinates associated with an illuminated object by using the illumination model. SPU **232** executes instructions from CPU **212** to produce sound signals that are output on an audio device (not shown). GPU **214** executes rendering instructions from CPU **212** and VPU **213** to produce images for display on a display device (not shown). That is, GPU **214**, using the texture coordinates and other illumination terms generated by VPU **213**, and rendering instructions from CPU **212**, renders the illuminated object in an image.

[0029] **FIG. 3** illustrates a modified environment reflection projection method stored in main memory **210** of **FIG. 2**, according to the present invention. In one embodiment of

the invention, the modified reflection projection method is used by the illumination model. In another embodiment of the invention, the modified reflection projection method is incorporated into the illumination model. **FIG. 3** includes an object's surface **305**, a point P on the surface **305**, a normal vector n to the surface **305** at point P (also referred to as a surface normal vector), an observer **310***a*, an observation vector e directed from the point P to observer **310***a*, and an x-axis **320**. In one embodiment of the invention, surface **305** is composed of polygon primitives (not shown), and at each vertex of each polygon, a point P is specified on surface **305**. For future reference, the z-axis (not shown) is perpendicular to x-axis **320** and is in the plane of **FIG. 3**, and the y-axis (not shown) is perpendicular to x-axis **320** and the z-axis and is out of the plane of **FIG. 3**. For simplicity of illustration, the **FIG. 3** embodiment of surface **305** is a line, however, any point P on any two-dimensional surface is within the scope of the invention. For example, the **FIG. 3** embodiment of surface **305** may be the intersection of a two-dimensional surface (not shown) with the x-z plane, and thus the normal vector n may have a vector component n_y along the y-axis.

[0030] According to one embodiment of the invention, the modified environment reflection projection method uses a modified version of a standard reflection formula to calculate a reflection vector r for each point P on surface **305**. The method then processes the reflection vector r to generate texture coordinates (s,t) for each point P. The standard reflection formula is r=e−2(e·n)n. For each point P specified on surface **305** with a given normal n, the standard reflection formula gives a reflection vector r based upon a given observer position specified by the observation vector e. The standard reflection formula is a vector relationship that satisfies Snell's law of reflection, where the angle of incidence a_1 (**FIG. 3**) is equal to the angle of reflection a_r (**FIG. 3**).

[0031] According to one embodiment of the invention, VPU **213** (**FIG. 2**) uses a modified version of the standard reflection formula to compute reflection vectors. For the point P on surface **305** located directly in front of observer **310***a* (i.e., observation vector e intersects x-axis **320** at a right angle), the standard reflection formula is modified such that a reflection vector r' given by the modified reflection formula is equal to the normal vector n. That is, r'=n. Thus, the modified reflection projection method produces the same result as the direct normal projection method when the point P on surface **305** is located directly in front of observer **310***a*.

[0032] In order to modify the standard reflection formula, a pseudo-normal vector n' is defined that bisects the angle of incidence a_1 subtended by the observation vector e and the normal vector n. That is, n'=(e+n)/(|e+n |) where |e+n | is the magnitude of e+n, and angle b_i is equal to angle b_r. When the pseudo-normal vector n' is substituted for the normal vector n in the standard reflection formula, the resultant modified reflection vector r' is equal to the normal vector n, since the modified reflection formula is based on the principle of Snell's law, where angle b_1=angle b_r. Thus, the modified reflection formula is expressed as r'=n=e−2(e·n')n'.

[0033] A simplification of the modified reflection formula is straightforward. Assuming that point P is located at (x,y,z)=(0,0,z), then the unit normal vector n has components [nx, ny, nz] and the unit observation vector e has components [0,0,−1], where brackets [] are used to specify vector quantities. For example, [nx, ny, nz] is another way of writing vector n. Now, substituting the components of n and e into the expression for n', one obtains n'=(e+n)/(|e+n|)=[nx, ny, nz−1]/(sqrt($nx^2+ny^2+(nz-1)^2$)). Expanding the argument of the square root in the denominator, one obtains $nx^2+ny^2+(nz-1)^2=nx^2+ny^2+nz^2+1-2nz=1+1-2nz=2(1-nz)$, since the normal vector n is a unit vector of magnitude one. If k=1/sqrt(2(1−nz)), then n'=k[nx, ny, nz−1].

[0034] Now, substituting n' into the modified reflection formula, one obtains r'=n=e−2(e·n')n'=e−2k^2(e·[nx, ny, nz−1])[nx, ny, nz−1]=e−(e·[nx, ny, nz−1])[nx, ny, nz−1]/(1−nz). That is, r'=e−(e·[nx, ny, nz−1])[nx, ny, nz−1]/(1−nz) for any given observation vector e. In other words, the modified reflection formula is valid for any given observation vector e=[ex, ey, ez], and any point P on surface **305** with an associated unit normal vector n=[nx, ny, nz]. For example, if observer **310***b* views point P along an observation vector e", then VPU **213** uses the modified reflection formula to compute a reflection vector r", where

[0035] r"=e"−(e"·[nx, ny, nz−1])[nx, ny, nz−1]/(1−nz).

[0036] The modified reflection formula may be simplified further, and expressed by a more compact mathematical relationship. For example, if the unit observation vector e=[0,0,−1] is relabeled as a constant reference observation vector e_o, then the modified reflection formula may be written as r'=e−(e·[nx, ny, nz−1])[nx, ny, nz−1]/(1−nz)=e−(e·(n+e_o))(n+e_o)/(1−nz).

[0037] **FIG. 4** is a flowchart of method steps for displaying an image of a reflective object based upon texture coordinates, according to one embodiment of the invention. In step **405**, a user loads video software into memory **210** (**FIG. 2**) via optical disc control unit **234** (**FIG. 2**), for example, and CPU **212** (**FIG. 2**) executes the video software. The video software may be an interactive or non-interactive video, and in an exemplary embodiment of the invention, the video software is a video game. In step **410**, CPU **212** generates rendering instructions for all reflective objects of a video frame. The rendering instructions may be generated in response to user input received via controller interface **220** (**FIG. 2**). In step **415**, VPU **213** (**FIG. 2**) executes the rendering instructions using an illumination model, and generates reflection vectors for each reflective object of the video frame. For example, a reflection vector is generated for each vertex point P (**FIG. 3**) of surface **305** (**FIG. 3**). Step **415** is further discussed below in conjunction with **FIG. 5**.

[0038] In step **420**, VPU **213** transforms the reflection vectors associated with each object to texture coordinates. The transformation may be a reflection vector mapping method, or may be configured using other known methods in the art. In addition, VPU **213** may compute a texture map composed of the texture coordinates for each reflective object. Next, in step **425**, VPU **213** sends the texture coordinates and/or texture maps to GPU **214**. In step **430**, GPU **214** prepares an image of each reflective object for display on a display device (not shown), based in part on the texture coordinates or texture map associated with each reflective object. GPU **214** may use other illumination terms generated by VPU **213** or CPU **212** in conjunction with the texture coordinates to prepare each reflective object for display as an image.

[0039] Next, in step **435**, CPU **212** determines if execution of the video game has been terminated. If execution has not terminated, then the method continues with the next video frame at step **410**. However, if in step **435**, CPU **212** ascertains that execution of the video game has terminated, then the method ends.

[0040] **FIG. 5** is a flowchart of method steps for generating reflection vectors for a reflective object, according to one embodiment of the invention. In step **505**, VPU **213** (**FIG. 2**) selects a vertex point P (**FIG. 3**) on surface **305** (**FIG. 3**) of the reflective object. Next, in step **510**, VPU **213** obtains vector components [nx,ny,nz] of a normal vector n (**FIG. 3**) to the surface **305** of the reflective object at the selected vertex point P. In one embodiment of the invention, the vector components are stored in registers (not shown) associated with CPU **212**. In another embodiment of the invention, the vector components are stored in a memory (not shown) associated with VPU **213**.

[0041] In step **515**, VPU **213** determines components [ex",ey",ez"] of the observation vector e" (**FIG. 3**), for example, directed from the point P to observer **310***b* (**FIG. 3**). VPU **213** may compute the components [ex",ey",ez"] or may receive the components from VPU **213** memory (not shown) or CPU **212** registers (not shown). Next, VPU **213** uses a modified reflection formula, the components (ex", ey",ez") of the observation vector e", and the vector components (nx,ny,nz) of the normal vector n to compute a reflection vector r" (**FIG. 3**), in step **520**. In step **525**, VPU **213** determines whether a reflection vector for each vertex point P associated with the reflective object has been computed. If a reflection vector for each vertex point P has not been computed, then in step **530**, VPU **213** selects another vertex point P on surface **305** of the reflective object. The method then continues at step **510**. However, if in step **525**, a reflection vector for each vertex point P has been computed, then the method ends.

[0042] The invention has been explained above with reference to several embodiments. Other embodiments will be apparent to those skilled in the art in light of this disclosure. The present invention may readily be implemented using configurations other than those described in the embodiments above. For example, the modified environment reflection projection method, according to the invention, may be executed in part or in whole by CPU **212**, VPU **213**, GPU **214**, or a rendering engine (not shown). Or, for example, the modified environment reflection projection method may be implemented in parallel by a multiprocessor system. Additionally, the present invention may effectively be used in conjunction with systems other than those described in the embodiments above. Therefore, these and other variations upon the disclosed embodiments are intended to be covered by the present invention, which is limited only by the appended claims.

What is claimed is:

1. A method for environment mapping, comprising the steps of:

determining a surface normal vector n at a point P on a surface of a reflective object;

determining an observation vector e from the point P to an observer; and

using a modified reflection formula to compute a reflection vector r based on the surface normal vector n and the observation vector e, the modified reflection formula based on reflection about a pseudo-normal vector n' at the point P on the surface.

2. The method of claim 1, wherein the pseudo-normal vector n' bisects an angle subtended by the surface normal vector n and a reference observation vector e_o, the reference observation vector e_o directed from the point P to the observer located directly in front of the point P.

3. The method of claim 1, wherein the modified reflection formula is r=e−(e·[nx, ny, nz−1])[nx, ny, nz−1]/(1−nz), where nx, ny, and nz are the components of the surface normal vector n.

4. The method of claim 1, wherein the point P on the surface of the reflective object is located at a vertex of a polygon.

5. The method of claim 1, further comprising the step of transforming the reflection vector r to texture coordinates (s,t).

6. The method of claim 5, further comprising the step of rendering the reflective object based on the texture coordinates (s,t) associated with each point P on the surface of the reflective object.

7. An electronic-readable medium having embodied thereon a program, the program being executable by a machine to perform method steps for environment mapping, the method steps comprising:

determining a surface normal vector n at a point P on a surface of a reflective object;

determining an observation vector e from the point P to an observer; and

using a modified reflection formula to compute a reflection vector r based on the surface normal vector n and the observation vector e, the modified reflection formula based on reflection about a pseudo-normal vector n' at the point P on the surface.

8. The electronic-readable medium of claim 7, wherein the pseudo-normal vector n' bisects an angle subtended by the surface normal vector n and a reference observation vector e_o, the reference observation vector e_o directed from the point P to the observer located directly in front of the point P.

9. The electronic-readable medium of claim 7, wherein the modified reflection formula is r=e−(e·[nx, ny, nz−1])[nx, ny, nz−1]/(1−nz), where nx, ny, and nz are the components of the surface normal vector n.

10. The electronic-readable medium of claim 7, wherein the point P on the surface of the reflective object is located at a vertex of a polygon.

11. The electronic-readable medium of claim 7, further comprising the step of transforming the reflection vector r to texture coordinates (s,t).

12. The electronic-readable medium of claim 11, further comprising the step of rendering the reflective object based on the texture coordinates (s,t) associated with each point P on the surface of the reflective object.

13. A system for environment mapping of a reflective object, comprising:

a memory configured to store a modified reflection model, the modified reflection model based on reflection about pseudo-normal vectors located at points on a surface of the reflective object;

a vector processing unit configured to compute reflection vectors using the modified reflection model; and

a graphics processing unit configured to render the reflective object in an image, the quality of the image dependent upon the computed reflection vectors.

14. The system of claim 13, wherein the modified reflection formula is $r=e-(e\cdot[nx, ny, nz-1])[nx, ny, nz-1]/(1-nz)$, where nx, ny, and nz are components of a surface normal vector n at a point P on the reflective surface, and e is an observation vector directed from the point P to an observer.

15. The system of claim 13, wherein a pseudo-normal vector n' bisects an angle subtended by a surface normal vector n at a point P on the reflective surface and a reference observation vector e_o, the reference observation vector e_o directed from the point P to an observer located directly in front of the point P.

16. The system of claim 13, wherein the vector processing unit is further configured to process each computed reflection vector to generate texture coordinates (s,t).

17. The system of claim 16, wherein the graphics processing unit uses the texture coordinates to render the reflective object as an image.

18. A system for environment mapping, comprising:

means for determining a surface normal vector n at a point P on a surface of a reflective object;

means for determining an observation vector e from the point P to an observer; and

means for using a modified reflection formula to compute a reflection vector r based on the surface normal vector n and the observation vector e, the modified reflection formula based on reflection about a pseudo-normal vector n' at the point P on the surface.

* * * * *